Enhancing Planning Heuristic with Landmarks

Jingjing Zhao[1,2], Dayou Liu[1]
1 College of Computer Science and Technology, Jilin University, Changchun, China
2 College of Business Administration, Changchun University of Technology, Changchun, China
zhaojingjing06@mails.jlu.edu.cn, dyliu@jlu.edu.cn

Yongming Yang[3]
3 Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science, Changchun, China
yangym@ciomp.ac.cn

***Abstract*—Recently, landmarks count heuristic can increase the number of problem instances solved and improve the quality of the solutions in satisfying non-optimal planning. In order to make landmarks count heuristic optimal, we give the solution to the overestimate of landmarks count heuristic. We extend landmarks count heuristic without action cost assignments, and prove that the extension of heuristic is admissible. Our empirical evaluation shows that the extension of heuristic is admissible and can be competed with the state-of-the-art of heuristic.**

***Index Terms*—artificial intelligence, intelligent planning, planning with landmarks, planning heuristic**

I. Introduction

Landmarks are facts that must be true at some point during the execution of any solution plan [1]. For example, clear(B) is a landmark for the task where the goal is to have block B stacked on block C, with another block A stacked on block B initially. Deciding landmark and finding the orderings between two landmarks are both PSPACE-complete [2]. Still, there are polynomial time algorithms for discovering and ordering landmarks such as the approach proposed by Hoffmann based on relaxed planning graphs [2], and the approach proposed by Richter based on domain transition graphs [3].

Currently, landmarks have been used in different types of planners. The roles which landmarks act as are different, such as preconditions sorting [4], guiding search [5], and planning heuristic estimators [6]. Landmarks and their orderings are extremely helpful in guiding the search for a plan. In particular, LAMA planner [7], the winner of the Sequential Satisfying Track at the 2008 International Planning Competition, utilizes such a landmarks-based heuristic, which is landmarks count heuristic. This technique uses the number of landmarks to estimate the goal distance of a state. As a result, this technique improves success rate and reduces the length of the generated plans. At the other hand, this heuristic is not optimal within a satisfying heuristic search.

In order to derive admissible heuristic estimates for optimal planning from a set of landmarks, Earpas and Domshlak propose cost-optimal heuristic with landmarks [8]. This approach allocates the cost of each action to the landmarks occurring in the action's effects. The core of this admissible heuristic is the cost assignment equations. Different assignment techniques cause different results. As long as the assignment technique follows the equations, the landmark heuristic which adopts the assignment technique is admissible. At the other hand, it takes times to compute the assignment costs and the choice of assignment techniques is another deliberate decision.

In this work, we depart landmarks from cost assignments and consider planning landmarks without cost assignments for admissible heuristic. We extend landmark count heuristic to make it admissible. And then the proof of this admissible heuristic is presented. Our empirical evaluation shows that this extension of heuristic has better performance in certain planning domains.

II. Notation and Background

We consider planning in the SAS+ planning formalism [9] which can be automatically generated from its PDDL description [10, 11]. An SAS+ planning task is a tuple (V, s_0, G, A). $V = \{v_1, ..., v_n\}$ is a set of state variables, each associated with a finite domain $dom(v_i)$. A fact is a pair $< v, d >$ (an assignment $v = d$) with $v \in V$ and $d \in dom(v)$. The union of the variable domains $F = \bigcup_i dom(v_i)$ is the set of facts. A state s is a complete assignment defined on all variables V. We use the function notation $s(v) = d$ and set notation $(v, d) \in s$ interchangeably. s_0 is an initial state, and the goal G is partial assignment to V. A is a finite set of actions, where each action is a pair (pre, eff) of partial assignments to V, called preconditions and effects, respectively. An action a is applicable in a state s if

This work is supported by the National Natural Science Foundation of China under Grant No.60773099, 60973088, 60803102, 60603030, 60873149, National High-Tech. R&D Program under Grant No.2006AA10Z245, 2006AA10A309. Corresponding author is Dayou Liu.

doi:10.4304/jcp.6.12.2590-2595

$pre \subseteq s$, produces the result state s' with $s'(v) = eff(v)$ where $eff(v)$ is defined and $s'(v) = s(v)$ otherwise. We write $s[a]$ for s' . In a similar way, we write $s[\pi]$ with $\pi =< a_1, ..., a_n >$ as an abbreviation of $s[a_1]...[a_n]$ if each action is applicable in the respective state. The action sequence π is a plan if $G \subseteq s_0[\pi]$.

We use the data structure Landmarks Graph for deriving heuristic. The landmarks and the orderings among them form Landmarks Graph. Most practical methods for finding landmarks are incomplete or unsound [1, 2, 3, 12, 13]. In what follows, we assume access to a complete and sound procedure; particularly, in our proof. We use LAMA's landmark discovery procedure, introduced by Richter et al in the experiments [3].

There are three types of orderings between landmarks in Landmarks Graph: natural ordering, necessary ordering, and greedy-necessary ordering. Let A and B be facts of an SAS+ planning task. If B is true at time i, A is true at some time j<i in each action sequence, there is a natural ordering between A and B, written A→B. If B is added at time i, A is added at time i-1 in each action sequence, there is a necessary ordering between A and B, written $A \rightarrow_n B$. If B is first added at time i , A is true at time i-1 in each action sequence, there is a greedy-necessary ordering between A and B, written $A \rightarrow_{gn} B$. Hoffmann et al. introduce another two types of orderings: reasonable ordering and obedient reasonable ordering, which are less important to our work. The most representative methods for computing these orderings utilize relaxed planning graph or domain transition graph. We use both of them to derived orderings [7].

III. PREVIOUS HEURISTIC WITH LANDMARKS COUNT

The most straightforward way of landmarks count heuristic is to estimate the goal distance from current state without using the number of actions for performing. Instead, it utilizes the number of landmarks to estimate the goal distance. The heuristic believes the assumption: each landmark is reached by an action and landmarks must be achieved by any plan. The number of actions need to be performed, approximately equals the number of landmarks in set l that still need to be achieved from current state onwards. l is estimated to be $n-(m-k)$, where n is the set of landmarks, m is the set of landmarks that have been accepted, and k is the set of accepted landmarks that are required again.

It is not hard to verify that the estimate is not admissible. For instance, in a BLOCKWORLD task, the goal is {hand-empty, on(A, B)}. n ={clear(A), clear(B), hand-empty, holding(A), on(A,B)}, m ={clear(A), clear(B), hand-empty, holding(A)}, k={hand-empty}, l = {hand-empty, on(A,B)}. While the equation $|l| = 2$ holds, it is possible a single action stack(A, B) reaches the goal from current state.

Landmarks count heuristic is not admissible, is caused by three approximate processes. The first approximation is that the number of landmarks left to reach is used to replace the number of actions left to perform. The second approximation is that the total number of n is obtained by incomplete or unsound practical methods. So the number of m is not accurate either, because m is computed from n . The third approximation is that k is computed by incomplete methods which only consider greedy-necessary orderings. Below, we show that the gap between the estimate and admissibility is not that hard to close.

IV. ENHANCING LANDMARKS COUNT HEURISTIC

A. Theories

In this part, we contribute to extend landmarks count heuristic to make it admissible without considering other elements except landmarks.

The first approximation is the key of landmarks count heuristic. However, the number of actions that need to perform is not always equal to the number of landmarks left to reach. In some tasks, the number of actions is greater than the number of landmarks. In another tasks, it is opposition. Thus, the first approximation needs enhancing. If the number of landmarks left to reach is less than or equals the number of actions that need to perform, the heuristic is admissible. If the number of landmarks left to reach is greater than the number of actions that need to perform, the heuristic fails to be admissible. This condition takes place, when there is at least one action to perform which makes two or more new landmarks be true at the same time. Thus, how to estimate the number of actions that need to perform more accurately, according to the set of landmarks is the key problem of enhancing landmarks count heuristic.

In order to solve this problem, there are another two questions need to answer. One is how to deal with the actions. Each action of them adds several landmarks at the same time. The other is how to deal with the landmarks. Each landmark of them can be added by several different actions. These two questions are answered by two admissible cost assignment equations in literature [8] for an admissible heuristic with landmarks. However, these equations give the range too wide. Sometimes the induced action cost partition adopted by some problem that obeys the admissible equations can be sub-optimal. Thus, it is necessary to limit the admissible equations more definitely. Or another way is provided.

Whatever relationships between landmarks and actions are, one landmark is added by one action at each time step. We call this action is the correspondence action of this landmark. When there are some actions that add the same one landmark, we choose the action which adds the largest number of new landmarks as the correspondence action of this landmark. New landmarks are the landmarks which belong to the set $n-m$. We update the correspondence action of other landmark that is added by this action with this action. Last, we utilize the number of correspondence actions of the landmarks in the set

$n-m$ as the enhancing landmarks count heuristic estimator. We do not compute the number of correspondence actions of the landmarks in the set $l = n-m+k$, because the landmarks of the set k are mostly the byproducts for adding landmarks of the set $n-m$ and usually k is 0 or very small.

The second approximation is caused by the practical methods. When landmarks count heuristic is proposed, there is no guarantee that the generated landmarks are complete. However, there is guarantee that the generated landmarks are sound with rapid speed. Although complete algorithm for finding landmarks allows the heuristic to be more accurate, it costs much time for large tasks. Complete algorithm fits for small tasks.

The third approximation is caused by unsound ordering between landmarks. The types of orderings between two landmarks are few and there are no guarantees of soundness for them. So k is computed by only considering greedy-necessary orderings which are more accurate than any other orderings. Although k is not accurate, it does not affect the admission of heuristic. It is because the smaller k is, the smaller l is.

B. Algorithm Specification

In this part, we give the admissible heuristic estimate, named enhancing landmarks count heuristic, as shown in (1). The symbol L represents certain landmark of the set $n-m$. The symbol a_L represents the correspondence action for the landmark L. The landmark L chooses the action a_L as its correspondence action because the effect of a_L contains the largest number of landmarks in the set n_m. At the same time, these landmarks added by a_L have the same correspondence action, except the landmark L' which have another action a' as its correspondence action because action a' adds more landmarks of the set n_m than a_L. Equation (1) considers the number of actions in the union set of a_L for all landmarks of the set $n-m$.

$$h^E = \Big| \bigcup_{L \in n-m} a_L \Big|. \tag{1}$$

Overall, our algorithm for computing enhancing landmarks count heuristic works as specified in Fig. 1. With what was said above, the algorithm should be self-explanatory. The input parameter “state” represents current state. The symbol “lgraph” is the data structure Landmarks Graph. This algorithm uses the functions “get_reached_landmarks”, which are defined in the class “State”, for getting the landmark set m. The data structure, the class and the function are the techniques from LAMA planner.

```
Compute_Enhancing_Heuristic(state)
{       reached_lms_num = state.get_reached_landmarks(reached_lms);
        n_m = lgraph.nodes - reached_lms;
        i=0;
        for each action a in A
        {       i++;
                ta=0;
                for each <v_i, d_i> in a.eff
                {    if <v_i, d_i> ∈ lgraph.nodes and <v_i, d_i> ∈ n_m,
                         then ta++;
                              operator_affect_lms[i].insert(<v_i, d_i>);
                }
                operator_ta.push_back(ta);
                if ta>0
                then for each LandmarkNode L in operator_affect_lms[i]
                        {    if(L.ta < ta)
                                then L.operator_index = i;
                                     L.ta = ta;
                        }
        }
        for each LandmarkNode L in n_m
        {    if L.ta >0
                then r.insert(L.operator_index);
        }
        h = r.size();
        return h;
}
```

Figure 1. Enhancing landmarks count heuristic.

C. Admissibility Proof

In this part, we prove the enhancing landmarks count heuristic admissible. The proof uses reduction to absurdity. Firstly, the assumption "enhancing landmarks count heuristic is not admissible" is provided. Then, a contradiction is inferred. Thus, the assumption does not hold. Proposition proves correct.

Proposition 1: If the generated landmarks and orderings are complete and sound, enhancing landmarks count heuristic is admissible.

Proof sketch is as follows. Assume enhancing landmarks count heuristic h^E is not admissible. That is the estimate $h^E = |\bigcup_{L \in n-m} a_L|$ of the state s in a planning task (V, s_0, G, A) is greater than the action number $h^*(s)$ of the best action path $\pi = < a_1, ..., a_p >$ from s to the goal G with $G \subseteq s[\pi]$. There are two types of relationships between the number of actions in path π and the number of landmarks in n. They are $|\pi| \geq |n|$ and $|\pi| < |n|$. If $|\pi| \geq |n|$ holds, there is no doubt that $h^E(s) \leq h^*(s)$ holds, because one landmark is at most added by one action at each time step with $h^E = |\bigcup_{L \in n-m} a_L| \leq n-m$. Only need to consider the condition $|\pi| < |n|$. The relationship between each action a of π and landmarks may be 1:0, 1:1 or 1:n. For the condition $|\pi| < |n|$, the case is that some actions add more than one landmark. For the set of landmarks, h^E obtains the least number of actions for achieving them. If the generated landmarks and orderings are complete and sound, the landmarks left to reach are in $n-m+k$. For $n-m \subseteq n-m+k$, $|\bigcup_{L \in n-m} a_L|$ is less than or equal to $|\bigcup_{L \in n-m+k} a_L|$. Thus the action number of π is greater than h^E. That is $h^E(s) \leq h^*(s)$ which contradicts with the assumption. The assumption fails. The above proposition proves correct.

V. Experimental Evaluation

To evaluate the proposed algorithm, we implemented our admissible heuristic procedure on the infrastructure of the Fast Downward planner [10], and used the landmark discovery techniques of LAMA [3]. We conducted an empirical study on a wide sample of planning domains from the international planning competitions. All experiments were run on 2.2GHz Intel T7500 CPU; the running time and memory limits were 30 minutes and 3GB respectively. The reported times do not include the PDDL to SAS+ translation as it is common to all planners. Table I depicts the results obtained over

TABLE I.
RUNTIMES OF THE PLANNERS ACROSS THE TEST DOMAINS

task	C*	h^E		h^{LM}	
		time(s)	nodes	time(s)	nodes
BLOCKWORLD domain					
4-0	6	0	7	0	10
4-2	6	0	7	0.01	9
6-0	12	0.01	13	0.01	18
6-1	10	0	11	0	28
8-2	16	0	17	0.01	37
DEPOTS domain					
1	10	0	18	0	11
3	27	0.42	1227	177.96	71334
7	21	0.01	72	0.13	456
10	24	0.44	1055		
13	25	0.07	158		
SATELLITE domain					
1	9	0	15	0	14
3	11	0	40	0.01	46
4	17	11.04	50673	23.19	49190
LOGISTICS domain					
4-0	20	0	21	0	23
4-1	19	0	20	0	23
4-2	15	0	16	0	18
5-0	27	0	28	0.01	30
5-1	17	0	18	0	22
5-2	8	0	9	0	9
6-0	25	0	26	0.01	29
6-1	14	0	15	0	15
6-2	25	0	26	0.01	28
6-9	24	0.01	25	0	33
7-0	36	0.01	37	0.01	49
7-1	44	0.01	45	0.01	80
8-0	31	0	32	0	38
8-1	44	0.01	45	0.01	67
9-0	36	0.01	37	0.01	52
9-1	30	0.01	31	0.01	32
10-0	45	0.86	3040	0.02	108
10-1	42	0.01	43	0.01	57
11-1	54	0.05	61	0.04	157
12-0	42	0.03	44	0.02	63
12-1	68	0.05	71	0.04	154

BLOCKWORLD, DEPOTS, SATELLITE and LOGISTICS domains.

Table I lists only tasks that have been well solved by one of the planners. Empty entries in the table denote tasks that were not solved by the respective technique with optimal solution. Column task denotes problem instance, column C* denotes optimal solution length. Column time denotes run time, and the other column denotes the number of expanded nodes of different heuristics and search procedures.

First, note that h^E outperforms the heuristic h^{LM} in some samples of planning domains. In LOGISTICS domain, h^E estimates the plan with the least nodes, although landmarks count heuristic estimates the plan with little nodes. That is because h^E guides the search along the optimal plan with greedy best-first search. This property is seldom seen before. In BLOCKWORLD domain, some experimental results are similar. Heuristic h^E works well for the structures of these problems. These problems have no goal which exists in their initial state. The effect of one action is the precondition of next action along the plan. The dependencies between actions are strong. However, there are differences between BLOCKWORLD domain and LOGISTICS domain. The actions of BLOCKWORLD domain mostly add several landmarks, while the actions of LOGISTICS mostly add one landmark. Although the actions of BLOCKWORLD add several landmarks at the same time, only one landmark of the set n_m is added. So the conditions of actions in both two domains are similar. The problems of BLOCKWORLD domain which h^E does not work well have goals in their initial states. The final goals are all obtained by destroying the goals in the initial states. We can make h^E adapt to this condition by considering landmarks needed again. That is the future research work.

The number of problems for SATELLITE and DEPOT domains which h^E works not well is greater than that of BLOCKWORLD and LOGISTICS domain. The reason is that h^E underestimates the number of actions left to execute for the complex relationships among these actions. The dependencies between actions are not one to one. The paths to the goals are so many that h^E can not discriminate. Or the h^E estimators of these paths are the same. The algorithm should visit all the nodes with the same estimators. Thus the time and the nodes expanded are much more than other domains. It is also the defect of landmark count heuristics.

As shown in Fig. 2, we compare the heuristic values provided by h^E and h^{LM} to the initial states of all tasks in BLOCKWORLD domain. Each point in Fig. 2 represents the initial state of a single task, with its x and y coordinates denoting the estimates provided to the initial state by h^E and h^{LM}, respectively. For many of these tasks, the cost of the optimal solution is currently not even known. It is not hard to see from the plots that the estimator of h^{LM} is greater than h^E. Sometimes the value of h^{LM} is greater than the length of optimal plan. So h^{LM} is not optimal estimator. On the other hand, the greater estimator is, more accurate the heuristic is. Thus, it will be much better to find one heuristic estimator which is admissible and much more close to optimal goal distances than h^E. This is another research work for the future.

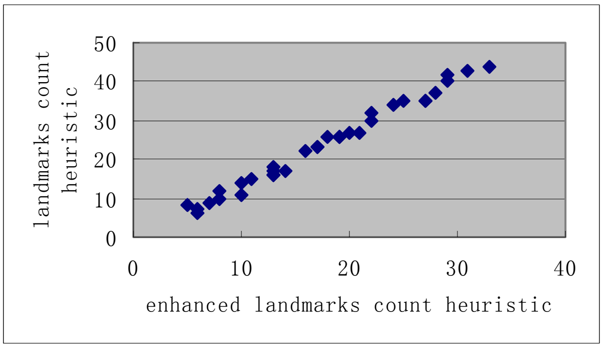

Figure 2. Estimates of h^E and h^{LM} on the initial states of all tasks from BLOCKWORLD domain.

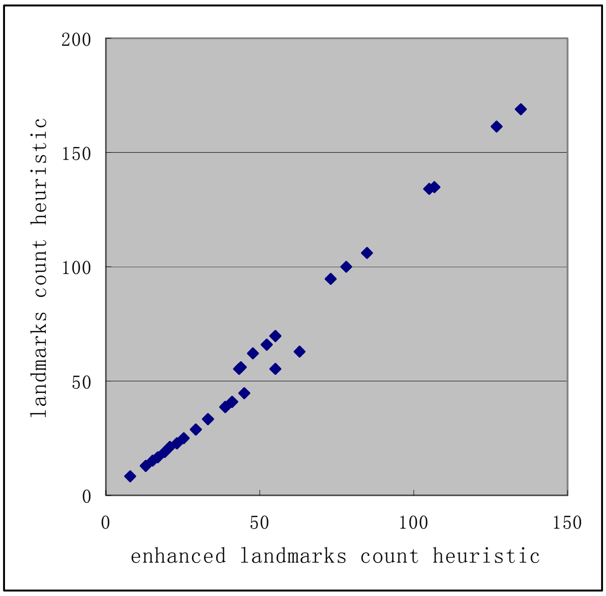

Figure 3. Estimates of h^E and h^{LM} on the initial states of all tasks from LOGISTICS domain.

As shown in Fig. 3, we compare the heuristic values provided by h^E and h^{LM} to the initial states of all tasks in LOGISTICS domain. Each point in Fig. 3 represents the initial state of a single task, with its x and y coordinates denoting the estimates provided to the initial state by h^E and h^{LM}, respectively. It is clear that the estimators of both two heuristic are the same for those smaller tasks. As the goal distance grows, h^{LM} quickly becomes larger than h^E. This is the same condition as Fig. 2.

VI. Conclusion

In this paper, we show that landmarks count heuristic can be extended to be admissible without considering complex cost assignments. The more important is that the admissibility proof is provided. The empirical evaluation indicates that the extension of heuristic is admissible. In particular the heuristic works well in the domains which have strong dependences between actions and clear structures. Furthermore, the properties obtained from the evaluation are encouraging that we can make further exploration of this idea in the context of landmark heuristics for the future research work.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No.60773099, 60973088, 60803102, 60603030, 60873149, National High-Tech. R&D Program under Grant No.2006AA10Z245, 2006AA10A309.

References

[1] J. Porteous, L. Sebastia, and J. Hoffmann, "On the extraction, ordering, and usage of landmarks in planning," in Proc. of European Conference on Planning, Toledo, Spain, 2001, pp.37-48.
[2] J. Hoffmann, J. Porteous, and S. Cresswell, "Ordered landmarks in planning," Journal of Artificial Intelligence Research, vol.22, pp.215-278, 2004.
[3] S. Richter, M. Helmert, and M. Westphal, "Landmarks revisited," in Proc. of Twenty-Third AAAI Conference on Artificial Intelligence, Chicago, Illinois, July 2008, pp.975-982.
[4] Dunbo Cai, Jorg Hoffmann, and Malt Helmert, "Enhancing the context-enhanced additive heuristic with precedence constraints," in Proc. of ICAPS, Thessaloniki, Greece, September 2009, pp.50-57.
[5] Blai Bonet and Malte Helmert, "Strengthening landmark heuristics via hitting sets," in Proc. of ECAI, Lisbon, Portugal, August 2010, pp.329-334.
[6] Malte Helmert and Carmel Domshlak, "Landmarks, critical paths and abstractions: what's the difference anyway," in Proc. of ICAPS, Thessaloniki, Greece, September 2009, pp.162-169.
[7] S. Richter and M. Westphal, "The lama planner: guiding cost-based anytime planning with landmarks," Journal of Artificial Intelligence Research, 2010, in press.
[8] E. Karpas and C. Domshlak, "Cost-optimal planning with landmarks," in Proc. of IJCAI, Pasadena, California, USA, July 2009, pp.1728-1733.
[9] C. Backstrom and B. Nebel, "Complexity results for sas+ planning," Computational Intelligence, vol.11 (4), pp.625-655, 1993.
[10] M. Helmert, "Concise finit-domain representations for PDDL planning task," Artificial Intelligence, vol.173, pp.503-535, 2009.
[11] M. Helmert, "The fast downward planning system," Journal of Artificial Intelligence Research, vol.26, pp.191-246, 2006.
[12] Emil Keyder, Silvia Richter, and Malte Helmert, "Sound and complete landmarks for and/or graphs," in Proc. of ECAI, Lisbon, Portugal, August 2010, pp.329-334.
[13] L. Zhu and R. Givan, "Landmark extraction via planning graph propagation," in Proc. of ICAPS, Trento, Italy, June 2003, pp.156-160.

Jingjing Zhao is born in 1981. She is a Ph.D. candidate of Jilin University at Changchun city in Jilin Province China. From Sept. 1999 to Jul. 2003, she was Computer Science and Technology major at Northeast Normal University. And she got B.E degree in 2003. After finished her bachelor, she pursued Computer Software and Theories studies for M.S degree at Northeast Normal University. She got her M.S degree in 2006. She is a Ph.D. student in the field of Artificial Intelligence under the supervision of Prof. Dayou Liu. Her main research interests include automated reasoning and intelligent planning.

Dayou Liu is born in 1942. He got M.S degree at Changchun city in Jilin Province China in 1981. He is a professor and doctoral supervisor at Jilin University, and is the senior member of China Computer Federation. He is the minister of Division of Information Sciences at Jilin University. His main research interests include uncertainty reasoning, temporal-spatial reasoning, knowledge engineering and expert systems, distributed artificial intelligence, etc.

Yongming Yang is born in 1980. He works at Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science, Changchun, China. In 2003, he got his B.E degree, majored in Computer Science and Technology at Northeast Normal University. In 2006, he got his M.S degree at Northeast Normal University, majored in Computer Application Technology. In 2009, he got PH.D. degree at Jilin University. His main research interests include Multi-agent Collaboration, Swarm Robotics.