Dynamic Classification Improved Ant Colony Optimization Algorithm and Its Applications (Postprint)

Authors: Chen Jia, You Xiaoming, Liu Sheng, Li Juan

Date: 2018-05-20T00:00:00+00:00

Abstract

To address the issues of slow convergence speed and imperfect cooperative mechanism in ant algorithms for robot path planning, a dynamically hierarchical improved ant algorithm is proposed by integrating the ideas of the wolf pack algorithm. First, a population hierarchical model is established based on a dynamic hierarchical operator. Then, to improve the cooperative mechanism, inter-population communication is increased through an alpha wolf influence strategy combined with roulette wheel selection. Finally, to enhance convergence speed, while retaining the pheromone update formula of the ant algorithm, a dynamically normalized pheromone update strategy is adopted to reflect the elitist effect. To validate the effectiveness, the grid method is employed to model the robot' s motion space, the proposed algorithm is applied to solving path planning problems, and it is compared with several other intelligent algorithms. Simulation results demonstrate that the algorithm exhibits faster convergence speed, can find the optimal path with relatively fewer iterations, and achieves higher efficiency.

Full Text

Preamble

Dynamic Hierarchical Improved Ant Algorithm and Its Application Research

Chen Jia , You Xiaoming †, Liu Sheng , Li Juan
(College of Electronic & Electrical Engineering; School of Management, Shanghai University of Engineering Science, Shanghai 201620, China)

Abstract: To address the slow convergence speed and inadequate coordination mechanisms of ant algorithms in robot path planning problems, this paper

proposes a dynamic hierarchical improved ant algorithm that incorporates concepts from the wolf pack algorithm. First, a population hierarchical model is established based on a dynamic hierarchical operator. Then, to improve the coordination mechanism, a head wolf influence strategy combined with roulette wheel selection is introduced to enhance inter-population communication. Finally, to accelerate convergence speed, a normalized dynamic pheromone update strategy is adopted while retaining the ant algorithm' s pheromone update formula to reflect the role of elites. To verify its effectiveness, the grid method is employed to model the robot' s motion space, and the proposed algorithm is applied to path planning problems and compared with several other intelligent algorithms. Simulation results demonstrate that the algorithm converges rapidly and can find optimal paths with relatively few iterations, achieving high efficiency.

Keywords: ant algorithm; wolf pack algorithm; dynamic hierarchical operator; head wolf influence strategy; dynamic pheromone update strategy

0 Introduction

Mobile robot path planning refers to the problem of finding an optimal collision-free path from a given start point to a target point in an obstacle-filled workspace, enabling the robot to reach its destination via the shortest possible distance while avoiding obstacles [1-2]. Research by Tian Zijian [3] and Shen Bowen [4] demonstrates that path planning technology remains a focal point in mobile robotics, with new and efficient improved algorithms continuing to be a central research priority [5].

Zhu Daqi et al. [6] categorize mobile robot path planning technologies into four types: template matching, artificial potential field, map-based, and artificial intelligence approaches. Among these, artificial intelligence-based path planning has gained widespread adoption due to its adaptive characteristics [7]. For instance, Wang Lei et al. [8] proposed an improved genetic algorithm that automatically adjusts crossover and mutation probabilities based on individual fitness. Huang Chen et al. [9] introduced a dynamic feedback A* ant colony algorithm based on closed-loop feedback principles. Jin Feihu et al. [10] developed an adaptive ant colony algorithm that adjusts parameters based on spatial obstacle distances and iteration counts.

As described above, artificial intelligence algorithms have achieved significant success and can be broadly classified into traditional and heuristic techniques. Heuristic techniques demonstrate greater environmental adaptability and can accelerate search speed, with numerous bio-inspired algorithms employing this approach. Among them, the Ant System (AS), first proposed by Dorigo in 1996, is an optimization algorithm that mimics the social division of labor and cooperative foraging behavior of ants [11]. While AS exhibits robustness and strong environmental adaptability, it requires considerable time for pheromone

accumulation. To improve algorithmic efficiency, various enhancements have emerged. Stutzle et al. [12] proposed the Max-Min Ant System (MMAS), which overcomes algorithm stagnation by limiting pheromone values, thereby improving diversity. Dorigo et al. [13] introduced the Ant Colony System (ACS), which adds a global pheromone update mechanism to local updates, enhancing convergence performance.

In addition to these improvements, Dorigo et al. [11] also proposed the Elitist Ant System (EAS), which employs a hierarchical approach where ants are classified into different ranks, and different pheromone update strategies are executed according to rank, emphasizing the role of optimal paths. Bullnheimer et al. [14] proposed the Rank-Based Ant System (RAS), another form of hierarchy that differs from EAS. While EAS divides the ant colony into elite and ordinary ants for stepwise pheromone updates, RAS employs only elite ants for pheromone updates through a ranking mechanism. Experimental results demonstrate that hierarchical mechanisms applied to ant systems yield good results, improving convergence speed and finding quality solutions in relatively short time. Inspired by these findings, other algorithms incorporating hierarchical mechanisms have gradually attracted attention.

Natural wolf packs are renowned for their social hierarchy and cooperative division of labor. Based on these group characteristics, Liu et al. [15] proposed the Wolf Colony Algorithm (WCA) in 2011, which divides wolves into different ranks with distinct responsibilities to improve algorithmic efficiency. Wu Husheng et al. [16] further improved and expanded WCA in 2013, proposing a novel wolf pack algorithm that applies the intelligent group behaviors, update mechanisms, and head wolf generation rules of wolf packs to solve complex problems. While the wolf pack algorithm converges quickly, it tends to fall into local optima and suffers from poor diversity in later stages. To address these limitations, Xue Junjie et al. [17] proposed an intelligent wolf pack algorithm to improve adaptability, while Wang Tao et al. [18] developed a new wolf pack search algorithm for clustering problems with good results.

Drawing inspiration from the hierarchical mechanisms of wolf pack algorithms, Feng Xiang et al. [19] proposed a social group search theory that introduces decision factors to classify social groups, where individuals in different categories exhibit varying degrees of autonomous consciousness. Experiments show that this improvement enhances algorithmic convergence and effectiveness, demonstrating the value of hierarchical structures.

To address the issues of inadequate convergence and imperfect coordination mechanisms in ant colony algorithms, where individuals within the same iteration lack sufficient communication, this paper proposes an improved ant algorithm based on dynamic hierarchy. By introducing a dynamic hierarchical operator to classify ants into different ranks, communication among ants is achieved through a head wolf influence operator and a dynamic hierarchical pheromone update strategy, thereby improving convergence speed while enhancing path optimization accuracy.

1.1 Elitist Ant System (EAS)

Also proposed by Dorigo et al. [11], the Elitist Ant System modifies the pheromone update strategy based on the Ant System. The pheromone update rule for paths traversed by elite ants—the global optimal path in the current iteration—differs from that of ordinary ants, emphasizing the dominant position of elite ants in each iteration.

The pheromone update rule in EAS is calculated according to Formula (1):

MATH_0

where (i,j) represents the original pheromone quantity on the path, is the pheromone evaporation rate, Δ (i,j) is the pheromone increment from ordinary ants, and Δ *(i,j) is the pheromone increment from elite ants, with specific content shown in Formula (2):

MATH_1

where Δ *(i,j) is the pheromone increment from elite ants,* is a fixed parameter, L* is the length of the global optimal path, and e is the number of elite ants.

The Elitist Ant System emphasizes the importance of elite ants in pheromone updates. The pheromone increase on the global optimal path in each iteration is a multiple of that on other traversed paths, which can accelerate convergence in early stages. However, as pheromone accumulates, excessive buildup in later stages causes ants to select these nodes with excessively high probability, suppressing the diversity development of the ant colony.

1.2 Discrete Wolf Pack Algorithm for TSP Problems

For solving TSP problems, Wu Husheng et al. [20] proposed a discrete wolf pack algorithm that embodies division of labor and cooperation through intelligent behaviors such as wandering, summoning, and besieging, while reflecting the hierarchical nature of wolf packs through head wolf generation rules and pack update strategies.

In TSP, the connection between the head wolf and fierce wolves in each iteration is realized through summoning behavior, which is visually represented by modifying the wolves' paths. Summoning behavior involves randomly selecting a path segment of length bstep from the head wolf' s path and replacing the corresponding segment (starting or ending with the same index) in a fierce wolf' s path with the same length. Mapping is then applied to adjust the cities in the fierce wolf' s path, ensuring each city index appears only once. An example is illustrated below:

Head wolf path: MATH_2
Fierce wolf path: MATH_3

When bstep = 3, the replacement length is 3. The head wolf' s replacement segment is MATH_4 , and the fierce wolf' s path after replacement becomes MATH_5 . After mapping adjustment, the fierce wolf' s path becomes MATH_6 , based on the mapping relationship.

Summoning behavior preserves the individual experience of fierce wolves while demonstrating the guiding role of the head wolf. However, the classification is relatively simple and does not reflect differentiated treatment of wolves at different ranks by the head wolf.

1.3 Social Group Hierarchy Division Factor

Feng Xiang et al. [19] proposed a social group search algorithm inspired by animal division of labor and hierarchical systems. Based on functional roles, the entire group is divided into three categories: individuals acting according to their own will, individuals acting according to others' will, and individuals eliminated from the group due to passive behavior. Building upon this, a decision factor is introduced as an evaluation value for whether an individual' s autonomous action will is strong: a larger value indicates stronger individual autonomous will and greater influence on others, while a smaller value indicates weaker autonomous will, making the individual more susceptible to influence without affecting others.

The decision factor can be constructed in two ways. If the fitness function is proportional to individual excellence, the factor is defined by Formula (3); when the fitness function value is inversely proportional to individual excellence, the factor is defined by Formula (4):

MATH_7

where _i is the decision factor value of individual i, fit_max is the optimal fitness, and fit_i is the fitness of individual i. Based on predefined thresholds and individual decision factor values, the entire social group can be divided into four categories: leader, leader-follower, follower, and eliminator, facilitating separate processing of members at each rank.

---

2 Dynamic Hierarchical Improved Ant Algorithm

This algorithm incorporates a dynamic hierarchical operator into the ant colony algorithm and performs further optimization through hierarchical head wolf influence strategies and dynamic pheromone update strategies based on ant social ranks. These operators are introduced in the following sections.

2.1 Overall Framework

The overall computational method of the dynamic hierarchical adaptive ant colony algorithm is shown in Figure 1 [Figure 1: see original paper]. After all ants complete path construction, the algorithm first classifies the ants in the current iteration through the dynamic hierarchical operator. Following the head wolf influence strategy, a second classification is performed, and dynamic pheromone update strategies are applied based on this new classification. This information is retained as the basis for path construction in the next iteration. When termination conditions are met, the current optimal path and path length information are output.

2.2 Dynamic Hierarchical Operator

In nature, wolf packs are generally divided into four classes: the alpha wolf couple, wolves with status second only to the leader, the main components of the pack, and a fourth category of wolves too weak to hunt. Accordingly, Wu Husheng et al. [16] divided the wolf pack into four classes in their proposed algorithm: head wolf, scout wolf, fierce wolf, and weak wolf. Inspired by this hierarchical mechanism, Feng Xiang [19] also used social group hierarchy division factors for individual rank classification.

Drawing inspiration from these ideas, this paper proposes a dynamic hierarchical operator that classifies the entire population into four different ranks based on individual fitness.

The rank R_i of individual i is determined by the hierarchical factor _i. In path planning, individual excellence is determined by path length, with shorter paths considered more excellent. Under this premise, the hierarchical factor is defined by Formula (5):

MATH_8

where length_i is the length of the path traversed by individual i, and min_length is the shortest path length in the current iteration, with _i determined by the ratio of length_i to min_length.

As shown in Formula (6), based on the hierarchical factor _i and thresholds LT and OT, the entire ant population is divided into four classes:

MATH_9

When _i = 1, individual i' s rank R_i is defined as head. Such individuals are analogous to the head wolf, exerting a guiding influence on the entire population, with typically only one such individual existing.

When 1 < _i LT, the individual' s rank is defined as leader, where LT is the rank threshold for this class. Individuals in this class are relatively excellent in the population, less influenced by the head, and can largely preserve their own experience, potentially becoming the new head.

When LT < _i OT, the individual' s rank is defined as ordinary, where OT is the upper limit threshold for rank division. They can be analogized to fierce wolves, representing ordinary members and the main composition of the population. They preserve individual experience while being influenced by the head.

When _i > OT, the individual is defined as outer. These individuals are considered eliminators, will definitely be guided by the head, and play no role in pheromone updates.

The hierarchical operation forms the foundation for the head wolf influence strategy and pheromone update strategy, embodying the strict rank division from the wolf pack algorithm. Meanwhile, the number of individuals at each rank is not fixed but adjusts according to the conditions of each iteration.

2.3 Hierarchical Head Wolf Influence Strategy

The influence of the head on other rank members can be manifested in various ways. Inspired by Wu Husheng [20], this algorithm implements a direct path segment replacement head wolf influence strategy: by randomly selecting two intersection points between the head wolf' s path and other rank members' paths, path segmentation and segment replacement are performed, preserving individual experience while demonstrating the head' s influence.

The application of the head wolf influence strategy varies according to path intersection conditions:

a) Two or more intersections besides start and end points.
As shown in Figure 2 [Figure 2: see original paper], when there are two or more intersections between the head wolf' s path and other rank members' paths (excluding start and end points), intersection points A and B are randomly selected. After segmenting the paths connected by points A and B, the head wolf' s AB path segment replaces other members' AB segments, demonstrating the head' s guiding role while preserving other members' experience.

b) One or fewer intersections besides start and end points.
As shown in Figure 3 [Figure 3: see original paper], when there is only one or fewer intersections between the head wolf' s path and other paths (excluding start and end points), the head wolf influence strategy does not take effect.

To enhance algorithmic diversity, a roulette wheel mechanism is also applied in the head wolf influence strategy, with different threshold values for different rank members: outer individuals are definitely influenced by the head, ordinary individuals have a relatively high probability of being influenced, while leader individuals, as candidates for head, have a low probability of being influenced.

Figure 4 [Figure 4: see original paper] illustrates the roulette wheel probability distribution for three rank members being influenced by the head. Excellent individuals have a smaller probability of being influenced and a greater chance

to leverage their own advantages, while weaker individuals are more likely to follow the head and passively accept the head' s experience.

2.4 Hierarchical Dynamic Pheromone Update Strategy

The information value left by different ranks of ants varies. To effectively utilize pheromone from elite individuals and reduce interference from weak individuals, a hierarchical dynamic pheromone update strategy is employed. In the early pheromone update stage, due to sparse pheromone distribution, ants rely more on heuristic information for path finding, resulting in excessive outer members and almost no leader class. In the later pheromone update stage, pheromone on superior paths tends to saturate, preventing ants from falling into dead zones. At this point, almost all population members belong to ordinary and leader classes, with leader class members increasing as the algorithm progresses.

Figure 5 [Figure 5: see original paper] shows the distribution of ants across ranks during different algorithm phases. With a constant total ant population, outer-class ants are most numerous in early stages and continuously decrease as the algorithm proceeds, increasing the number of leader and ordinary ants. In later stages, the leader class contains the most ants.

To ensure that rank importance is consistently reflected throughout the algorithm and that guidance from higher ranks remains stable, this algorithm incorporates normalization processing while employing hierarchical dynamic pheromone update strategies. For path segments traversed by individual k, the pheromone is calculated according to Formula (7):

MATH_10

where (i,j) is the current pheromone quantity at node (i,j), is the pheromone evaporation rate, and Δ (i,j) is the pheromone increment.

Formula (8) defines the pheromone increment Δ (i,j), where Q , Q , Q , Q represent the total pheromone that can be released by each rank: head rank has the largest total pheromone, leader rank has the second largest, ordinary rank has less impact than the first two, and outer rank individuals have no impact on pheromone updates. n , n , n , n represent the number of members in each rank.

MATH_11

The hierarchical pheromone update strategy reflects the influence of different rank individuals. Experience from excellent individuals in the current iteration significantly impacts path selection in the next iteration, while experience from weak individuals, having less reference value, is not emphasized. This update strategy demonstrates the guiding role of elite individuals, improving algorithmic convergence. Moreover, because it depends on the dynamic hierarchical operator described above, it adapts to changing iteration conditions, avoiding algorithmic imbalance.

3 Experimental Results

To verify the algorithm' s effectiveness, this paper conducts comparative experiments between the proposed algorithm and the ant algorithm, ant colony system, and elitist ant algorithm across five different environments. The effectiveness of each operator is also validated through component-by-component comparisons with the original ant algorithm.

3.1 Overall Algorithm Performance Comparison

Table 1 presents the parameter settings for the four algorithms. AS and the proposed algorithm use only local pheromone evaporation factors, while ACS and EAS use both local and global pheromone evaporation factors. EAS requires setting the number of elite ants, whereas the proposed algorithm requires presetting pheromone intensities for different ranks and hierarchical thresholds.

After configuring the required parameters for each algorithm, performance comparison experiments are conducted in five different environments. Figure 6 [Figure 6: see original paper] compares the optimal paths found by the proposed algorithm and AS in the 50×50_1 environment, while Figure 7 [Figure 7: see original paper] compares the iteration processes of the four algorithms. The results show that both the proposed algorithm and EAS can find optimal solutions with faster convergence than the other three algorithms, demonstrating good overall performance.

Environment 40×40_1 is also a complex environment. As shown in Figures 8 [Figure 8: see original paper] and 9 [Figure 9: see original paper], the proposed algorithm, EAS, and ACS can find optimal solutions, while AS shows significant error in its optimal path. The proposed algorithm consistently finds optimal solutions with relatively minimal iterations, maintaining good overall performance.

Environment 20×20_1 is relatively simple compared to the above environments. All four algorithms can find optimal solutions. Figures 10 [Figure 10: see original paper] and 11 [Figure 11: see original paper] demonstrate that the proposed algorithm converges faster and finds quality solutions earlier than other algorithms, showing relatively better problem-solving performance.

Environment 50×50_2 is another complex scenario with numerous traps. Figures 12 [Figure 12: see original paper] and 13 [Figure 13: see original paper] show that the proposed algorithm and EAS can find optimal solutions, ACS approaches optimal solutions, while AS produces significantly inferior solutions. The proposed algorithm also requires fewer iterations to find optimal solutions, yielding good overall results.

Environment 40×40_2 is a complex multi-trap environment where algorithms easily fall into local optima, preventing them from finding optimal solutions.

As shown in Figures 14 [Figure 14: see original paper] and 15 [Figure 15: see original paper], AS performs worst in this environment, while ACS and EAS can find relatively good but still suboptimal solutions. The proposed algorithm demonstrates faster convergence and higher quality optimal solutions compared to other algorithms.

Table 2 compares the performance of the four algorithms across the five environments, with each algorithm run 10 times. The results show that EAS and the proposed algorithm exhibit high convergence rates, while EAS, ACS, and the proposed algorithm produce high-quality solutions. The proposed algorithm achieves optimal solutions in all five environments, particularly demonstrating its advantages in more complex scenarios.

3.2 Head Wolf Influence Strategy Effect Comparison

To evaluate the head wolf factor' s effectiveness, both algorithms are controlled with identical parameters and environments: one using the original AS algorithm, the other incorporating only the head wolf influence strategy.

As shown in Figures 16 [Figure 16: see original paper] and 17 [Figure 17: see original paper], the head wolf influence strategy proves particularly effective when path intersections are numerous. By replacing path segments, it increases traversal frequency of certain paths, boosting their pheromone concentration and improving algorithmic convergence while demonstrating inter-individual coordination and enhancing solution quality to some extent.

3.3 Dynamic Pheromone Update Strategy Effect Comparison

The proposed algorithm employs a hierarchical pheromone update mechanism where individuals of different ranks update pheromones using different strategies.

As shown in Figures 18 [Figure 18: see original paper] and 19 [Figure 19: see original paper], this mechanism effectively reflects the guiding role of excellent individuals on subsequent iterations, improving both solution quality and convergence speed, even functioning effectively in complex environments.

---

4 Conclusion

This paper proposes an improved ant colony algorithm based on dynamic hierarchy. Through a dynamic hierarchical factor, the entire population is divided into four classes, and different degrees of head wolf influence strategies and pheromone update strategies are executed according to individual ranks, thereby improving algorithmic convergence to some extent.

Experiments demonstrate that in simple environments with fewer traps, the proposed algorithm can find optimal paths with relatively few iterations, showing

favorable efficiency. However, solution quality for complex scenarios with numerous traps remains to be improved. Future research directions include further enhancing solution quality and investigating the relationship between algorithm parameters and map environments to improve environmental adaptability.

References

[1] Zuo Dali, Nie Qingbin, Zhang Liping, Ding Dukun. Research on ant colony optimization algorithm in mobile robot path planning [J]. Modern Manufacturing Engineering, 2017, 39(5): 44-48.
[2] Zhu Qingbao, Zhang Yulan. Ant colony algorithm for robot path planning based on grid method [J]. Robot, 2005, 27(2): 132-136.
[3] Tian Zijian, Gao Xuehao. Disaster relief robot path planning based on improved artificial potential field method [J]. Industry and Mine Automation, 2016, 42(9): 37-42.
[4] Shen Bowen, Yu Ningbo, Liu Jingtai. Intelligent scheduling and path planning for warehouse logistics robot swarms [J]. CAAI Transactions on Intelligent Systems, 2014, 9(6): 659-664.
[5] Yang Juncheng, Li Shuxia, Ceng Zengyu. Research and development of path planning algorithms [J]. Control Engineering, 2017, 24(7): 1473-1480.
[6] Zhu Daqi, Yan Mingzhong. Survey on mobile robot path planning technology [J]. Control and Decision, 2010, 25(7): 961-967.
[7] Wu ZongSheng, Fu WeiPing. A review of path planning method for mobile robot [J]. Advanced Materials Research, 2014, 3470(1030): 1588-1593.
[8] Wang Lei, Li Ming, Cai Jincao, et al. Application research of improved genetic algorithm in mobile robot path planning [J]. Mechanical Science and Technology, 2017, 36(5): 711-716.
[9] Huang Chen, Fei Jiyou, Liu Yang, et al. Smooth path planning method based on dynamic feedback A* ant colony algorithm [J]. Transactions of the Chinese Society for Agricultural Machinery, 2017, 48(04): 34-40+102.
[10] Jin Feihu, Gao Huijun, Zhong Xiaojian. Application of adaptive ant colony algorithm in space robot path planning [J]. Journal of Harbin Institute of Technology, 2010, 42(07): 1014-1018.
[11] Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents [J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, 1996, 26(1): 29-41.
[12] Stutzle T, Hoos H. H. MAX-MIN ant system [J]. Future Generation Computer Systems, 2000, 16(8): 889-914.
[13] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem [J]. IEEE Trans on Evolutionary Computation, 1997, 1(1): 53-66.
[14] Bullnheimer B, Hartl R F, Strauss C. A new rank based version of the ant system-a computational study [J]. Central European Journal of Operations Research, 1999, 7(1): 25.
[15] Liu CG, Yan XH, Liu CY, Wu H. The wolf colony algorithm and its application [J]. Chinese Journal of Electronics, 2011, 20(2): 212-216.

[16] Wu Husheng, Zhang Fengming, Wu Lushan. A new swarm intelligence algorithm—wolf pack algorithm [J]. Systems Engineering and Electronics, 2013, 35(11): 2430-2438.
[17] Xue Junjie, Wang Ying, Li Hao, et al. A wolf pack intelligent algorithm and its convergence analysis [J]. Control and Decision, 2016, 31(12): 2131-2139.
[18] Wang Tao, Wang Yong, Meng Liping. An improved wolf pack search algorithm and its application to solving clustering problems [J]. Computer Applications and Software, 2016, 33(12): 257-263.
[19] Feng Xiang, Ma Meiyi, Shi Yin, et al. Robot path planning based on social group search algorithm [J]. Journal of Computer Research and Development, 2013, 50(12): 2543-2553.
[20] Wu Husheng, Zhang Fengming, Li Hao, et al. Discrete wolf pack algorithm for solving TSP problem [J]. Control and Decision, 2015, 30(10): 1861-186.

Note: Figure translations are in progress. See original paper for figures.