Advanced Materials Research
ISSN: 1662-8985, Vols. 108-111, pp 1354-1359
doi:10.4028/www.scientific.net/AMR.108-111.1354
© 2010 The Author(s). Published by Trans Tech Publications Ltd, Switzerland.
Online: 2010-05-11

An Improved Ant Colony Optimization Supervised by PSO

Zhigang Zhou

Department of Computer Science and Technology, Dezhou University, China

zzg@dzu.edu.cn

Keywords: Ant Colony Optimization; Particle Swarm Optimization; continuous optimization.

Abstract. Combined with the idea of the particle swarm optimization (PSO) algorithm, the ant colony optimization (ACO) algorithm is presented to solve the well known traveling salesman problem (TSP). The core of this algorithm is using PSO to optimize the control parameters of ACO which consist of heuristic factor, pheromone evaporation coefficient and the threshold of stochastic selection, and applying ant colony system to routing. The new algorithm effectively overcomes the influence of control parameters of ACO and decreases the numbers of useless experiments, aiming to find the balance between exploiting the optimal solution and enlarging the search space.

Introduction

Russl Eberhart and James Kennedy [1] initially proposed the particle swarm optimization, PSO, as an approach for modeling inspired from social and cultural organization of animals and insects group, their aptitudes to organization and to solve problems. The PSO, is then proposed as an optimization technique [2][3].

Ant-based approaches was proposed for combinatorial optimization problems [4][5], originally motivated by the attempt to solve the well known Travelling Salesperson Problem (TSP), the inventors of the approach soon recognized that their technique is applicable to a much larger range of problems. In an explicit form, this insight was established by the creation of the Ant Colony Optimization (ACO) metaheuristic by Dorigo and Di Caro [6]. Gutjahr carried out a proof of ACO convergence; Sttzle and Dorigo, later on, gave evidence to the convergence of an optimal solution.

The analytical investigation of the runtime of ACO algorithms is still a very new issue at the moment, there are already some first results available, and they promise that a large amount of knowledge on this issue can be won within the next years. Some of the existing results already allow a comparison of specific ACO algorithms with counterparts from the field of Evolutionary Algorithms, in particular with the so-called Evolutionary Algorithms (EA).

In parallel to the development of ACO, a few years ago a new paradigm "PSO" ushered. At first it was simply an approach, yet now it has attracted the interest of researchers around the globe. PSO is intended to give an overview of important work that gave direction to research in particle swarms as well as some interesting new directions and applications.

Our objective matter that we will test consists of joining ACO and PSO to build a hybrid new approach for the TSP problem. As soon as one passes from the question of convergence of an algorithmic iteration to that of the speed of convergence, one cannot expect anymore to obtain very general results. This is a consequence of the so-called"no-free-lunch theorems" which basically state that for each algorithm that performs well on some optimization problems, there must necessarily be other problems where it fails to be efficient.

The aim of the present article is to introduce a new PSO ACO paradigm based on which results of the outlined type can be derived in a mathematically sound way, to present already available results, to give an introduction into the techniques by which these results have been obtained, and to outline their scope of application.

The paper is organized as follows: Section 2 introduces the social intelligence paradigm with a focus on social insects. Section 3 presents the PSO and ANT based optimization since these

This article is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license
(https://creativecommons.org/licenses/by/4.0)

techniques are the key issues of our proposal. In section 4 we detail the A.S.PSO, Ant Supervised by PSO algorithm. The paper is concluded in section 5.

Accomplishments of the Social Insects

An insect may have only a few hundred brain cells, but insect organizations are capable of architectural marvels, elaborate communication systems, and terrific resistance to the threats of nature. Extrapolating from Lorenz' descriptions of the fixed action patterns, the stereotypical" instinctive" behaviors of organisms, Wilson theorized that the dazzling accomplishments of ant societies could be explained and understood in terms of simple fixed action patterns, which, Wilson discovered, included behavioral responses to pheromones, chemicals that possess a kind of odor that can be detected by other ants.

The study of complex systems and the rise to prominence of computer simulation models of such systems gave scientists the tools they needed to model the simple behaviors of ants and how they could combine to produce an effect that is much more than the sum of its parts, and these insights have in turn led to more insights about the nature of man and society and about the physical world. Insect sociality is a classic example of the emergence of global effects from local interactions.

In fact, all their published models derive from the activities of the social insects.

PSO

The initial ideas on particle swarms of Kennedy (a social psychologist) and Eberhart (an electrical engineer) were essentially aimed at producing computational intelligence by exploiting simple analogues of social interaction, rather than purely individual cognitive abilities. The first simulations [1] were influenced by Heppner and Grenander's work and involved analogues of bird flocks searching for corn. These soon developed [1] into a powerful optimization method, Particle Swarm Optimization (PSO).

In PSO a number of simple entities, the particles, are placed in the search space of some problem or function, and each evaluates the objective function at its current location. Each particle then determines its movement through the search space by combining some aspect of the history of its own current and best (best-fitness) locations with those of one or more members of the swarm, with some random perturbations. The next iteration takes place after all particles have been moved. Eventually the swarm as a whole, like a flock of birds collectively foraging for food, is likely to move close to an optimum of the fitness function.

Each individual in the particle swarm is composed of three D-dimensional vectors, where D is the dimensionality of the search space. These are the current position x_i, the previous best position p_i, and the velocity v_i.

The current position x_i can be considered as a set of coordinates describing a point in space. On each iteration of the algorithm, the current position is evaluated as a problem solution. If that position is better than any that has been found so far, then the coordinates are stored in the second vector, p_i. The value of the best function result so far is stored in a variable that can be called $pbest_i$ (for "previous best"), for comparison on later iterations. The objective, of course, is to keep finding better positions and updating p_i and $pbest_i$. New points are chosen by adding v_i coordinates to x_i, and the algorithm operates by adjusting v_i, which can effectively be seen as a step size.

The particle swarm is more than just a collection of particles. A particle by itself has almost no power to solve any problem; progress occurs only when the particles interact.

$$V^i_{j,t+1} = WV^i_{j,t} + C_1R_1(P^i_{j,t} - X^i_{j,t}) + C_2R_2(P^{i,g}_{j,t} - X^i_{j,t}) \quad (1)$$

$$X^i_{j,t+1} = X^i_{j,t} + V^i_{j,t+1} \quad (2)$$

Where j = 1,...,n, i = 1,...,N, and are two positive constants, and are random values in range [0, 1] and

- W is the called inertia weight of the particle. This is employed to control the impact of previous history of velocities on the current velocity, thus to influence he trade-off between global and local exploration abilities of the particles. A larger inertia weight w facilitates global exploration while a smaller inertia weight tends to facilitate global exploration while a smaller inertia weight tends to facilitate local exploration to fine-tune the current search area. Suitable selection of the inertia weight w can provide a balance between global and local exploration abilities requiring less iteration for finding the optimum on average. A nonzero inertia weight introduces.

The preference for the particle to continue moving in the same direction as in the previous iteration.

- C_1R_1 and C_2R_2 are called control parameters. These two control parameters determine the type of trajectory the particle travels. If R_1 and R_2 are 0.0, it is obvious that v=v+0andx=x+v (for w = 1). It means the particles move linearly. If they are set to very small values, the trajectory of x rises and falls slowly over time.
- $P_t^{i,g}$ is the position of the global best particle in the population, which guides the particles to move towards the optimum, The important part in MOPSO is to determine the best global particle $P_t^{i,g}$ for each particle i of the population. In single objective PSO, the global best particle is determined easily by selecting the particle that has the best position. But in MOPSO , $P_t^{i,g}$ must be selected from the updated set of non-dominated solutions stored in the archive A_{t+1}. Selecting the best local guide is achieved in the function FindBestGlobal (A_{t+1} , X_t^i) for each particle i.
- P_t^i is the best position that particle i could find so far. This is like a memory for the particle i and keeps the non-dominated (best) position of the particle by comparing the new position X_{t+1}^i in the objective space with P_t^i (P_t^i is the last non-dominated (best) position of the particle i).

ACO

A metaheuristic that belongs now to the most prominent and most frequently applied techniques for search and heuristic optimization started its development fifteen years ago out of the seminal work by Dorigo et al.[4][5]: the ant-based approach to the solution of combinatorial optimization problems. Originally motivated by the attempt to solve the wellknown Travelling Salesperson Problem (TSP), the inventors of the approach soon recognized that their technique is applicable to a much larger range of problems. In an explicit form, this insight was established by the creation of the Ant Colony Optimization (ACO) metaheuristic by Dorigo and Di Caro [6]. In the meantime, there exist several hundreds of publications reporting on successful applications of ACO in a large variety of areas. For a recent survey, we refer the reader to the profound and comprehensive textbook by Dorigo and Sttzle [7].

Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni showed how a very simple pheromone-following behavior could be used to optimize the travelling salesman problem.

The travelling salesman problem (TSP) is a classical example of a combinatorial optimization problem, which has proved to be an NP-hard. In the TSP, the objective is to find the salesman's tour to visit all the N cities on his list once and only once, returning to the starting point after travelling the shortest possible distance. Additionally, we assume that the distance from city i to city j is the same as from city j to city i (symmetrical TSP). A tour can be represented as an ordered list of N cities. In this case, for N>2 there is N!/2N different tours (the same tour may be started from any city from among N cities and traversed either clockwise or anti-clockwise).

Procedure ACO

1. Initialize pheromone trails $\tau_{i,j}$ on the edges (i,j) of c;

2. For iterationm=1,2,...do

RETRACTED

For agents=1,...,S do
set i, the current position of the agent, equal to the start node of C;
set u, the current partial path of the agent, equal to the empty list;

3. While a feasible continuation (i, j) of the path u of the agent exists do
select successor node j with probability p_{ij} ,
where
p_{ij}=0, if continuation (i,j)is infeasible, and

$$p_{i,j} = \frac{g(\tau_{ij}, \eta_{ij}(u))}{\sum_{i,r} g(\tau_{ij}, \eta_{ij}(u))}$$

where the sum is over all feasible continuations (i, r), otherwise;
continue the current path u of the agent by adding edge (i, j) to u and setting i=j;
end while

4. Update the pheromone trails $\tau_{i,j}$;
5. End of Two For

- Where the values $\eta_{ij}(u)$ are called heuristic information values.
- Contrary to $\tau_{i,j}$, the quantities $\eta_{ij}(u)$ are allowed to depend on the entire partial path u traversed, as indicated by the argument u.
- The function g combines pheromone trail and heuristic information. The most popular choice is :

$$g(\tau, \eta) = \tau^{\alpha} \cdot \eta^{\beta} \quad (3)$$

with parameters $\alpha > 0$ (often chosen as $\alpha = 1$) and $\beta > 0$.
- The diverse ACO variants mainly differ in the way the update of the pheromone trails is performed. That we have make PSO supervising an ant optimizer called"Ant Supervised by PSO" (A. S. PSO) to solve continuous optimization problems. This issue can be seen in figure 1.

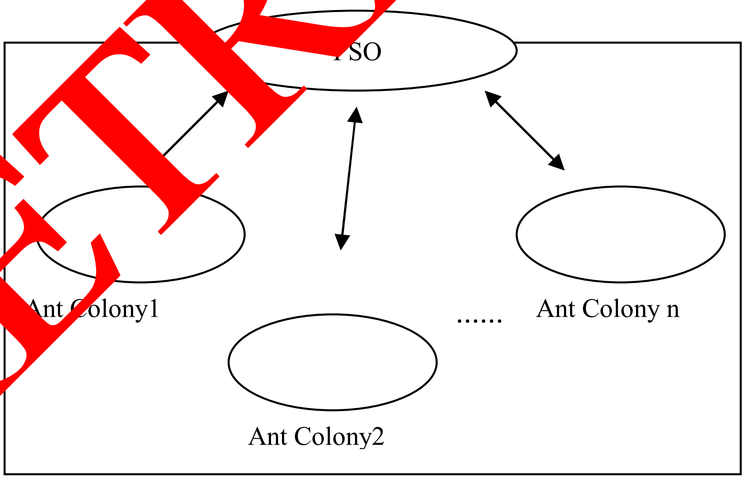

Fig. 1. Search space

The Ant Colony Supervised By PSO Algorithm (A.S.PSO)

Our proposal is to make PSO supervising an ant optimizer. In this paper we propose an Ant colony algorithms supervised by Particle Swarm Optimization to solve continuous optimization problems. Traditional ACO are used for discrete optimization while PSO is for continuous optimization problems. Separately, PSO and ACO shown great potential in solving a wide range of optimization problems.

Aimed at solving continuous problems effectively, this paper develops a novel ant algorithm "Ant Supervised by PSO" (A.S.PSO) the proposed algorithm can reduce the probability of being trapped in

RETRACTED

local optima and enhance the global search capability and accuracy. An elitist strategy is also employed to reserve the most valuable points. Pheromone deposit by the ants' mechanisms would be used by the PSO as a weight of its particles ensuring a better global search strategy. By using the A.S.PSO design method, ants supervised by PSO in the feasible domain can explore their chosen regions rapidly and efficiently.

1) Initialization particle swarm
2) Ant Initilization of spaces and their respective research
3) Run the Ant (iteration n) and find the best solution
4) To assimilate the best solutions to particles PSO
5) To iterate PSO

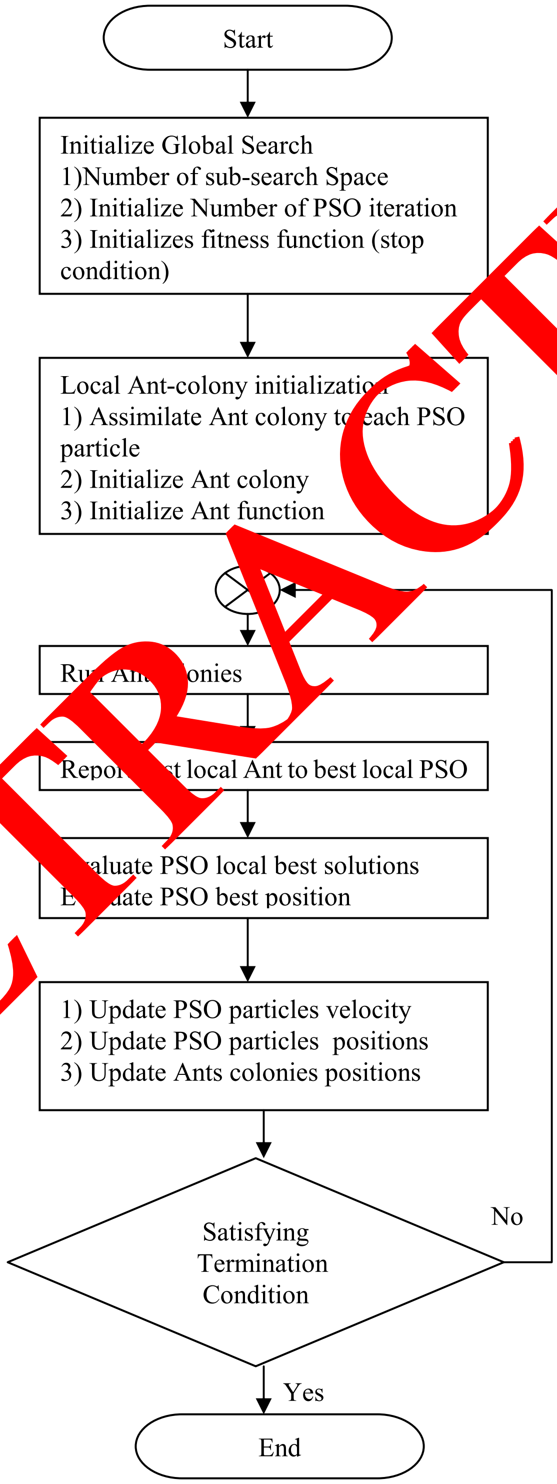

Fig. 2. Procedure A.S.PSO

RETRACTED

RETRACTED

First we start by initializing the search in light of research in space, the number of iterations for PSO and the cost function. The second step is to initialize the ant, the cost of this function and assimilate an ant for each PSO particle. Thereafter we proceed to simulate the ant algorithm and integrate the best solution to the ant than the PSO. Finally we will evaluate the best local solutions and the best position of the PSO. The last step is to change the speed and position of particles PSO, and the positions that of Ant. If conditions are satisfied we reach the end if not we go back to the third stage.

Since walk has symmetry characteristics, we assigned sub-Ant, respectively to the left foot and to the right one. Both must cooperate to ensure a generation of the joints trajectories ensuring a stable walking. The joins considered here are the hip, the knee and the ankle joints.

The inference graph of the particle swarm is conceived as follows: The memory particles are connected hierarchically from bottom to top, such a connection means that particle (p0.2), particle 2 of swarm 0 communicate it's position to both particles (p0.1) and (p0.3), it gathers also the respective positions' of these particles. Particle (p0.3) has only the position of (p0.2). P0 is a specific particle used to represents the center of mass of the body. This particle is critic since the analysis of its coordinates will help the decision on whether the generated joint position is a statically stable one or not C.

Conclusion

In this paper we introduced first the PSO and ACO algorithms then we developed our proposal called A.S.PSO; issued from a habitation of both techniques. It took the best of the ACO and to integrate the PSO. This proposal will be soon applied to solve classical test bench problems such as TSP; it will be used also to optimize the control of a humanoid robot.

The model of the locomotion system is to be extended to higher degree of freedom including a new and complicated particles representation called A.S.PSO. We will conduct more testes to evaluate and enhance The A.S.PSO proposal methodology.

A humans have an natural knowledge of the their bodies limits, even in the cases of amputation the human still assuming his leg or arm when this one is no longer part of the body. For its locomotion system, IZIman, will better include a virtual model with a limited degree of freedom, such a model is useful to help the robot to optimize path including different criteria such as global time and energy minimizing; as a human think, IZIman will carry on simulations to optimize his strategies choices.

References

[1] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," IEEE International Conference on neural Networks, pp. 1942–1948, 1995.

[2] J. Dreo, A. Petrowski and P. Siarray, "methaheuristiques pour l'optimisation difficile," Eyrolles, Paris, 2003.

[3] Y. Shi and R. C. Eberhart, "Fuzzy adaptive Particle swarm Optimisation," Congress on evolutionary computation, Seoul, 2001.

[4] M. Dorigo, V. Maniezzo and A. Colorni, "Positive feedback as a search strategy," Technical Report TR-POLI-91-016, Dipartimento di Elletronica, Politecnico di Milano, Milan, Italy, 1991.

[5] M. Dorigo, V. Maniezzo and A. Colorni, "Ant system: optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics, pp. 1–13, 1996.

[6] M. Dorigo and G. Di Caro, "The Ant colony optimization metaheuristic," In D. Corne, M. Dorigo and F. Glover (Eds.), New ideas in optimization Maidenhead: McGraw-Hill. pp. 11–32, 1999.

[7] M. Dorigo and T. Sttzle, "Ant colony optimization," Cambridge: MIT, 2004.