OPTIMAL LOCATION MANAGEMENT IN MOBILE COMPUTING WITH HYBRID GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION (GA-PSO)

Lipo Wang and Guanglin Si
School of Electrical and Electronic Engineering, Nanyang Technological University
Block S1, 50 Nanyang Avenue, Singapore 639798

ABSTRACT

Location management is an important and complex issue in mobile computing. In the reporting cell location management scheme, the concept of reporting cell was introduced to simplify the management process. As a result, there is a need to optimize the allocation design of reporting cells, since it determines the overall location management cost. We present an approach based on particle swarm optimization (PSO) to obtain minimum cost in the location management problem. Genetic algorithms (GA) are used to improve PSO. Simulation results show that the hybrid algorithm can locate the optimal solution in most cases, within a shorter computational time compared with other algorithms.

Index Terms**—** location management, mobile computing, particle swarm optimization, genetic algorithm

1. INTRODUCTION

Continuous expansion of mobile networks, despite restricted mobile spectrum, makes the reduction of overheads in communication a highly important issue. Considerable traffic is caused by the process of determining mobile terminal locations. Thus, it is necessary for the network system to track and predict the position of each mobile terminal, so that it can route an incoming call appropriately with minimum resource consumption. The strategy aiming to achieve that is referred as location management [1],[2].

The location management problem is important, since it has a direct influence on the quality of service (QoS) and thus user satisfaction. A poor strategy does not only waste the bandwidth resource of the wireless network, but also requires more energy on the terminal side, which is a critical issue for mobile devices.

On the other hand, the location management problem is complex, due to the variety in practical conditions. For example, some mobile terminal users may stay within a region for a long time, while some others travel a lot. A good location management scheme should meet both their needs with minimum cost. Thus for a mobile service provider, it is difficult to design a one-for-all location management strategy.

These facts provide the motivation for solving the location management problem. In this paper, we study optimal location management, i.e., minimization of the location management cost in term of overheads in mobile computing, using PSO. The reasons for choosing PSO are its simplicity and fast convergence [5]-[16].

Let us now review the location management problem in more details. In location management, we need to keep a record of user's location information, so that when a call for the user arrives, the network can route the incoming call to the appropriate station. It involves two operations, namely, location update and paging. Location update is the operation performed by a mobile device, to inform a nearby network station that the device is under the coverage of the station. Thus the incoming message for this device will be transmitted by the proper station later. Paging is to look for the requested terminal when an incoming call arrives. It is important to note that the paging operation is not limited to the station to which the device did the last location update. It is the location management strategy that determines whether a station should do paging for a certain incoming call.

Accurate location information can save resources spent on searching for the particular user (paging). However, to achieve that, more frequent location updates are required, which increases the updating cost. Therefore, an optimization problem in location management is about how to achieve the balance between location update and paging, so as to minimize the total cost of mobile communications.

The location area scheme is one of location management strategies that are commonly used in existing networks [1]. In this scheme, the whole network is partitioned into a few regions called location areas. The update strategy is that no location update operation should be performed, unless the user moves into a different location area. The reporting cell scheme is a location management strategy proposed recently [1]. According to this strategy, the whole network is separated by some special cells, called reporting cells. The location update strategy here is that update operation can be performed only in reporting cells, and it is only performed when the user enters a reporting cell, except that it is the one where he did his last location update. The reporting cell model will be used in the rest of this paper, since it has the potential to provide a better service quality compared to other schemes [1]. It also transforms the location management problem into a combinational optimization problem on the arrangement of reporting cells and thus optimization algorithms can be readily applied.

As mentioned above, location management involves two elementary operations, location update and paging. Therefore, the location management cost during a period of time T can be calculated by:

$$\text{Total Cost} = C * N_{LU} + N_P$$

where N_{LU} is the number of location updates performed during time T, N_P denotes the number of paging performed during time T, and C is a constant representing the cost ratio of location update and paging. It is recognized that the cost of a location update is usually several times higher than the cost of paging [3] and C =10 is used in this paper. We ignore any other network traffic for controlling purpose.

Further, N_{LU} and N_P can be calculated by:

$$N_P = \sum_{j=0}^{N-1} w_{cj} * v(j)$$

$$N_{LU} = \sum_{i \in S} w_{mj}$$

where W_{mj} stands for the movement weight associated with cell j, W_{cj} denotes the call arrival weight associated with cell j, V_j denotes the vicinity value of cell j, N denotes the total number of cells in the network, and S denotes the set of reporting cells in the network.

By combining above three equations, the following formula is obtained to calculate the overall location management cost of a particular configuration in the reporting cell scheme:

$$\text{Total Cost} = C * \sum_{j=0}^{N-1} w_{cj} * v(j) + \sum_{i \in S} w_{mj}$$

In order to simulate the allocation of reporting cells, each particle's location is represented by a binary number with N-bit length, where N is the size of the network. In this binary array, 1 stands for a reporting cell, while 0 denotes a non-reporting one.

Optimal location management in the reporting cell scheme was studied using three artificial life techniques, genetic algorithm, tabu search, and ant colony optimization (ACO) [2],[15].

2. HYBRID GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION (GA-PSO)

PSO often locates nearly optimal solutions at a fast convergence speed, but fails to adjust its velocity step size to continue optimization in the binary search space, which leads to premature convergence.

In contrast, research has shown that genetic algorithms (GA) can adjust its mutation step size dynamically in order to better reflect the granularity of the local search area. This indicates a potential to surpass the performance of PSO in term of solution quality. However, GA suffers from a slow convergence speed. Therefore, hybrid GA PSO has been proposed to overcome those problems and combine advantages of PSO and GA.

Besides the difference in individual representation, there are two main differences between GA and PSO, i.e., selection process and exploration direction. The function of selection in GA is to reallocate resources according to individuals' performance, letting those with promising results have more opportunity to survive. While in PSO, there is no such selection. However, its absence is partially offset by the concepts of global and personal bests. By combing the velocity and location update equations,

$$v_{id}(t+1) = v_{id}(t) + c_1 * \psi_1 * (p_{id}(t) - x_{id}(t)) + c_2 * \psi_2 * (p_{gd}(t) - x_{id}(t))$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)$$

we obtain

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t) + c_1 * \psi_1 * (p_{id}(t) - x_{id}(t)) + c_2 * \psi_2 * (p_{gd}(t) - x_{id}(t))$$

Hence if $x_{id}(t)$ is considered as a parent and $x_{id}(t+1)$ is treated as an offspring, it can be seen that the values of global and personal bests can affect the production of the next generation, playing a role similar to selection.

In fact, the concept of personal best also simplifies PSO, in the sense that no computational resource is spent on selection. However, as a tradeoff, memories are needed to keep a record of personal bests, which makes the population size of a particle swarm twice the number of particles, since each individual contains a distinct personal best plus its current location, while those in GA are only award of current values.

A second distinction between GA and PSO lies in the exploration direction, or their approaches to manipulation of individuals.

PSO uses highly directional operations. Each particle's velocity vector has a clear direction. When the personal best and current global best are given, the whole possible exploration area can be determined, although the specific new direction is further determined by two random numbers, C1 and C2.

In contrast, exploration direction GA is rather uncertain. Consequently, particle swarm optimization will perform best in environments where the average local gradient points towards the global optimum, but may have difficulties in finding optima when the average local gradient points in the wrong direction or is constantly changing. Thus, GA will be able to perform better in such an environment, since it does not have a strictly defined exploration area.

The idea of combining GA and PSO is not new [11]. However, most attempts so far have been focusing on introducing GA's mutation operation into PSO [16], aiming to overcome PSO's early convergence in local optima. Although the introduction of mutation operator has been proven to be helpful for PSO [16], it is not considered as a suitable choice for the location management problem, because in a high dimensional search space, a single mutation operation occurred on one dimension may not lead to significant change on the individual's fitness value. Furthermore, due to the small value of mutation rate, it is likely to take a long time to wait for an effective mutation to occur, which is considered inefficient. Therefore, in our proposed hybrid GA-PSO, the crossover operation is also included, which can improve the diversity of individuals.

Our hybrid GA-PSO algorithm is proposed as follows:

GA-PSO Pseudo–Code:

Initialization

Generate 2 groups of individuals. One (named as G) is for GA, and the other one (named as P) is for PSO. Each has a population size of 2N for an N-dimensional problem. Velocity V is also initialized at this stage for PSO.

Evaluation

Calculate fitness value for G and P.

Repeat

GA method

Apply GA operators (crossover and mutation) to the first group G. Then, rank the new population G_{new} on the basis of the fitness values

PSO method

Apply PSO operators (velocity and position updates) to the second group P. Then rank the new population P_{new} on the basis of the fitness values

Interchange and Update

Interchange a certain percentage (i.e., the hybrid ratio) of the worst individuals between G_{new} and P_{new}. Update corresponding fitness value

Until the termination criterion is reached.

In this paper, the constriction coefficient version of PSO is used, due to its better performance shown in the next section.

Our hybrid GA-PSO is explorative, with two new terms, i.e., hybrid frequency and hybrid rate. The functionalities of those two parameters are discussed below.

Hybrid frequency stands for the number of iterations between two hybrid operations. According to the experiment results, this determines the overall performance of GA-PSO. If the frequency is too high, both algorithms will get interrupted too often and consequently, convergence may be delayed. On the other hand, an extreme high frequency makes GA-PSO meaningless, since there will be no interaction between GA and PSO before one of termination conditions is met. Simulations were conducted to explore any dependencies of solution quality and runtime on population size and hybrid frequency.

Since the best or nearly best result is obtained in all cases, it is concluded that there is no specific dependency of solution quality on hybrid frequency or population size. Regarding the runtime, a good setting for the hybrid frequency is about 10, e.g., 7 to 9 for the 4x4 network and 13 to 15 for 6x6 network.

Hybrid ratio is the portion of individuals undergoing hybrid operations.

An important function for hybrid ratio is to balance the tradeoff between GA's global exploration abilities and PSO's local exploitation abilities. When a large ratio is applied, the resulted GA-PSO is more similar to PSO, in the sense that a fast convenience is likely to occur. On the other hand, with a small ratio applied, it performs more like GA. Simulation result has shown that a generally good range for hybrid rate is in [0.4, 0.5]. In the simulation later, it is set to 0.5, which means that the interchange should be between the worse 50% of population in both GA and PSO.

3. SIMULATIONS ON LOCATION MANAGEMENT PROBLEM WITH GA-PSO

In this section, results of GA-PSO simulations are presented and compared with other algorithms, including GA, tabu search, and ant colony optimization (ACO).

For the comparison purpose, the stopping condition used in this simulation follows examples in previous work. It is assumed that the calculation of the solution's fitness, will take most of the computation time. Further, since the fitness calculation is a common computation among different algorithms, the number of fitness calculation allowed is used as the stopping condition for the different algorithms [2],[14]. Each algorithm was run for 200 times [2],[14]. For the 4x4 and 6x6 networks, the differences between various algorithms are not significant and we thus present here the comparisons for the 8x8 network only.

Table 1. Result Comparisons with [2],[14] for the 8x8 network.

Method	Average Value	Minimum Value	Maximum Value
GA	14.005	13.782	14.671
TS	13.791	13.782	13.999
AC1	14.107	13.801	14.407
AC1,rank	13.860	13.782	14.007
AC1,tournament	13.868	13.782	14.051
AC2	14.393	13.901	15.373
AC2,rank	14.032	13.782	14.462
AC2,tournament	14.045	13.782	14.417
GA-PSO	13.829	13.782	13.947

The values of GA are from previous work on the same problem set [2],[14]. It can be seen that GA-PSO significantly out-performs GA and PSO, and is also better than most other algorithms, while similar to tabu search.

Three termination conditions are applied to the program: (1) for all algorithms, terminating the algorithm when the optimal result is obtained; (2) terminating when the result remains unchanged for a certain number of generations (10 for PSO, 20 for GA, and 5 hybrids for GA-PSO); and (3) terminating GA and GA-PSO after 500 iterations.

Table 2. Simulation results for the 8x8 network (we obtained these results by simulations, rather than taken from other papers).

Method	Average Value	Standard Deviation	Success Percentage
GA	15.283	0.5567	2.00%
PSO	16.062	0.53327	0%
GA-PSO	13.812	0.0869	65.50%

Table 2 shows that PSO does not provide acceptable results in most cases and tends to fall into a local minimum quickly. As discussed previously, that is due to the quick loss of diversity in its swarm. PSO's disadvantage in the combinatorial problem may also contribute to its failure, since it is harder for particles to escape form an existed converging point in a discrete search space [15].

Compared with PSO, GA's performance is much better in term of average value and success percentage. That indicates that GA has a higher chance to get the global optimal result, since it has an efficient exploration ability to search in the global scope of the search space. Meanwhile, its slow convergence speed is a reflection of its lack in an effective local search mechanism for accurately searching near a good solution.

As a combination of GA and PSO, GA-PSO actually combines an exploitation ability of PSO and an exploration ability of GA.

4. CONCLUSIONS

We showed that the original PSO failed to solve the location management problem. We proposed a new hybrid between GA and PSO, i.e., GA-PSO, in which individuals in GA and PSO are swapped. GA-PSO' performance has been compared with a few algorithms on the location management scenario and results have shown that GA-PSO performs better than most other algorithms, while performs similarly in comparison with tabu search.

REFERENCES

[1]. N.A. Bar and I. Kessler, "Tracking Mobile Users in Wireless Communications Networks," *IEEE Trans. Information Theory*, vol. 39, pp. 1877-1886, 1993.

[2]. R. Subrata and A.Y. Zomaya, "Location Management in Mobile Computing," *Proc. ACS/IEEE Int'l Conf. Computer Systems and Applications*, 2001.

[3]. T. Imielinski and B.R. Badrinath, "Querying Locations in Wireless Environments," *Proc. Wireless Comm. Future Directions*, 1992.

[4]. P.R.L. Gondim, "Genetic Algorithms and the Location Area Partitioning Problem in Cellular Networks," *Proc. IEEE 46th Vehicular Technology Conf.*, 1996.

[5]. A.P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, Wiley, 2005.

[6]. Shenheng Xu and Yahya Rahmat-Samii, "Boundary conditions in particle swarm optimization revisited", *IEEE Trans. Antenn. Propagat.,* vol. 55, no. 1, pp. 760-765, Mar. 2007.

[7]. Carlisle, A., Dozier, G., 2001. *An Off-the-shelf PSO*, Proceedings of the Workshop on Particle Swarm Optimization, p.1-6.

[8]. Kennedy J and Mendes R, "Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms", *SMCIA/03: Proceedings of the 2003 IEEE International Workshop on Soft Computing in Industrial Applications*, pages 45-50, 2003

[9]. R. Brits, A.P. Engelbrecht and F. Van Den Bergh, "Solving Systems of Unconstrained Equations using Particle Swarm Optimization", Submitted to *IEEE Conference on Systems, Man, and Cybernetics*, Tunisia, 2002.

[10]. El-Gallad, A. I., El-Hawary, M. E., and Sallam, A. A., "Swarming of intelligent particles for solving the nonlinear constrained optimization problem," *Engineering Intelligent Systems for Electrical engineering and Communications*, vol. 9, no. 3, pp. 155-163, Sept.2001.

[11] G. Mahinthakumar and M. Sayeed. "Hybrid genetic algorithm - local search approaches for groundwater source identification problems", *Special Issue on Evolutionary Computation, ASCE Journal of Water Resources Planning and Management*, 2004.

[12]. C.Z. Janikow, Z. Michalewicz, "An experimental comparison of binary and floating point representation in genetic algorithms", in *Proceedings of the International Conference on Genetic Algorithms*, San Diego, CA, USA, (1991), pages. 31–36.

[13]. P. Angeline, "Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences", In *V. W. Porto and et al., editors, Evolutionary Programming*, volume 1447 of Lecture Notes in Computer Science, pages 601–610. Springer, 1998.

[14]. Riky Subrata and Albert Y. Zomaya, "A Comparison of Three Artificial Life Techniques for Reporting Cell Planning in Mobile Computing", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 14, NO. 2, February 2003

[15]. L. Cagnina, S. Esquivel and R. Gallard, "Particle Swarm Optimization for Sequencing Problem: A Case Study", in *proc. of IEEE EC (2004),* pages 536–541.

[16]. Qian-Li Zhang, Xing Li, and Quang-Ahn Tran, "A modified particle swarm optimization algorithm", *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference* on Volume 5, Issue 18-21 Aug. 2005 Page(s): 2993 - 2995 Vol. 5