

Using Micro-Genetic Algorithms to Improve Localization in Wireless Sensor Networks

Vincent Tam, King-Yip Cheng and King-Shan Lui

Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong.

Email: {vtam, kycheng, kslui}@eee.hku.hk

Abstract— Wireless sensor networks are widely adopted in many location-sensitive applications including disaster management, environmental monitoring, military applications where the precise estimation of each node position is inevitably important when the absolute positions of a relatively small portion as anchor nodes of the underlying network were predetermined. Intrinsically, localization is an unconstrained optimization problem based on various distance/path measures. Most of the existing localization methods focus on using different heuristic-based or mathematical techniques to increase the precision in position estimation. However, there were recent studies showing that nature-inspired algorithms like the ant-based or genetic algorithms can effectively solve many complex optimization problems. In this paper, we propose to adapt an evolutionary approach, namely a micro-genetic algorithm, as a post-optimizer into some existing localization methods such as the Ad-hoc Positioning System (APS) to further improve the accuracy of their position estimation. Obviously, our proposed MGA is highly adaptable and easily integrated into other localization methods. Furthermore, the remarkable improvements attained by our proposed MGA on both isotropic and anisotropic topologies of our simulation tests prompt for several interesting directions for further investigation.

Index Terms—wireless sensor networks, localization, optimization techniques, heuristic search methods, micro-genetic algorithms, distance measures.

I. INTRODUCTION

A wireless sensor network is a network consisting of thousands of sensors that span a large geographical region. These sensors are able to communicate with each other to collaboratively detect objects, collect information, and transmit messages. Sensor networks have become an important technology especially for environmental monitoring, military applications, disaster management, etc [5, 11]. However, as sensors are usually small in size, they have many physical limitations. For example, due to its limited size, a sensor does not have a very powerful CPU and is limited in computational

power and memory. On the other hand, a sensor is powered by a battery instead of a power outlet. This limitation in energy puts extra constraints in the operations of sensors. As recharging is difficult, sensors should smartly utilize its limited energy in collecting, processing, and transmitting information.

In many applications, sensors have to know their geographical locations. Theoretically, Global Positioning System (GPS) can be used for a sensor to locate itself. In reality, it is not practical to use GPS in every sensor node because a sensor network consists of thousands of nodes and GPS becomes very costly. To solve the problem, many localization methods have been developed. Instead of requiring every node to have GPS installed, all localization methods assume only a few nodes (≥ 3) are equipped with GPS hardware. These nodes are called *anchor* or *beacon* nodes and they know their positions without communicating with other nodes. Other normal sensors then obtain distance information through talking to each other and derive their positions based on the information. A good localization protocol should reduce the error in position estimation by using reasonable number of messages. The computation and memory required should be limited as well due to the nature of sensor networks mentioned above.

Most of the existing works focus on increasing the accuracy in position estimation by using different mathematical techniques such as triangulation [10], multilateration [12], multidimensional scaling [7, 13, 14], convex optimization [3], etc. In these methods, information provided by every anchor node is used, without considering the position of that anchor. To the best of our knowledge, there is only limited research on studying the significance of anchor node selection, particularly for localization algorithms in anisotropic¹ sensor networks. In a previous work [2], we showed that the accuracy of a localization algorithm could be increased by wisely selecting a subset of anchor nodes rather than all anchor nodes when applying the Ad-hoc Positioning System (APS) [10], a distributed and hop-by-hop localization algorithm, to estimate node positions in anisotropic sensor networks [8]. Basically, anisotropic

Based on "Improving Localization in Wireless Sensor Networks with An Evolutionary Algorithm", by Vincent Tam, King-Yip Cheng, and K.S. Lui which appeared in the Proceedings of the IEEE Consumer Communications and Networking Conference 2006, Las Vegas, U.S.A., January 2006. © 2006 IEEE.

¹ A network is isotropic if measurements in all directions are exhibiting the same properties; otherwise, it is anisotropic.

sensor networks possess challenging properties to certain localization algorithms such as the APS due to various limiting factors including the geographical shape of the involved region, different node density, irregular radio patterns, etc. In such challenging anisotropic cases, we identified the criteria of “good” anchor nodes and developed an algorithm for selecting these nodes. Clearly, our previous proposal of selecting “good” anchor nodes can be readily extended to other general sensor networks.

After all, we regard localization as intrinsically an unconstrained optimization problems (UOPs) [1, 20] for which the aim is to minimize the total estimation error of computed positions for all the sensor nodes considered with respect to their true positions. It is worth noting that there are other approaches that always assume the existence of 3 or more sufficiently close neighbors fulfilling certain bounding conditions. An example is the linear programming approach [6] that formulates localization as a linear program with a system of linear constraints to deduce all the node positions through their sufficiently close neighbors. Here, in this paper, we tend to relax such restrictive bounding condition(s), and thus adopt a more generalized formulation of the localization problem in any wireless sensor network defined as follows.

$$\min \sum_{\forall i \in NA} \left(\sqrt{ep_i^2 - rp_i^2} \right) \quad (1)$$

where NA is the set of “non-anchor” sensor nodes, ep_i is the estimated position of node $i \in NA$, and rp_i is the true position of node i . Basically, different localization methods like the APS choose different anchors and accordingly their associated distance measures to a particular sensor node i to compute its estimated position ep_i so as to minimize its estimation error. With a network size of hundreds or even thousands of sensor nodes with 10% as anchors, searching for the “best” k anchor nodes, where $k \geq 3$, among the 10% anchors can be readily computationally expensive. For instance, with a network of 200 sensor nodes and 10% as anchors, the number of possible combinations of anchors needed to be considered for all 180 non-anchor nodes is $180 \times {}_{20}C_3 = 205,200$ in each iteration. From our previous empirical observation, when implemented with an exhaustive search mechanism to look for the best 3 anchors, a simple localization algorithm like the APS will become computationally intensive, thus taking almost a day to run on a Pentium IV desktop PC installed with a 3 GHz processor for a network of 200 sensor nodes with 10% as anchors. Clearly, using the exhaustive search to choose the best 3 anchors seems to be an infeasible approach to reduce the estimation error of most existing localization methods. In this regard, since the arbitrarily chosen anchors may not be able to provide the most “valuable” distance measures for minimizing the overall estimation error, we propose to take the estimated positions computed by most existing localization methods for further improvement by efficient, possibly heuristic-based, post-optimizers.

In the past decade, there were many challenging constrained or unconstrained optimization problems [20]

found in real-life applications [9, 20] efficiently solved by many nature-inspired algorithms including the ant-based algorithms [18], evolutionary algorithms (EAs) [4, 19] or swarm intelligence (SI) [15]. Therefore, in this paper, we consider to further enhance our previously improved APS technique using an adapted micro-genetic algorithm (MGA) [16, 19], with a relatively smaller population size (often ≤ 30) to focus its search effort on limited resources such as the number of iterations allowed, as a post-optimizer into other existing localization methods [3, 10, 13, 14]. Our adapted MGA for localization consists of two major genetic operators, namely the descend-based mutation and crossover operators, trying to improve on the objective values of individual chromosomes, with each denoting a position estimation of the current node of interest until the best k chromosomes converged to the same solution or the maximum number of iterations exceeded. To demonstrate the feasibility of our proposal, we implemented a prototype of our adapted MGA, and evaluated the performance of the improved APS integrated with our adapted MGA against that of the improved APS using simulations on the challenging anisotropic networks. When compared to the nonlinear optimizer provided in the Matlab Optimization Toolbox, our adapted MGA-based optimizer achieved remarkable improvements on both isotropic and anisotropic topologies of our simulation tests, prompting for more detailed analysis in our future investigation. In addition, our proposal of MGA-based post-optimizer is so generic that it can readily be integrated into other heuristic-based or mathematical localization methods.

This paper is organized as follows. Section II reviews some preliminary concepts including our previous study [2] on the significance of anchor node selection in the APS for localization in anisotropic sensor networks. In Section III, we consider the original micro-genetic algorithm for solving complex constrained optimization problems, and also our adapted MGAs as post-optimizers to further improve the accuracy of localization methods. Section IV summarizes the experimental results of the improved APS integrated with our adapted MGAs against those of the improved APS only. Lastly, we conclude our work in Section V.

II. RELATED WORK

In this section, we firstly review the original APS algorithm [10] based on the triangulation technique used in GPS to estimate the positions of other sensor nodes with reference to the fixed positions of anchor nodes dispersed inside any general sensor network. Later, we will consider our previous proposal [2] to improve the position estimation through more careful anchor selection in such challenging cases of anisotropic sensor networks.

A. Ad-Hoc Positioning System (APS)

APS [10] is one of the earliest methods developed for localization. Based on the triangulation used in GPS, APS is a distributed protocol that requires reasonable computation, memory, and message overhead. APS

assumes there are at least three anchor nodes in a sensor network. Each normal sensor tries to find out its distance to the anchor nodes. When the distance information to three or more anchor nodes is obtained, the sensor node can compute its own position using triangulation. The key question APS answers is how a node finds out its distances to the anchors. Three methods are described: *DV-hop*, *DV-distance*, and “*Euclidean*” *propagation*. Among them, *DV-hop* and *DV-distance* receive most attention. Both *DV-hop* and *DV-distance* measure distance in a hop-by-hop manner. Each sensor is required to communicate with its immediate neighbors only. In *DV-distance*, an anchor node A^* starts out by sending a broadcasting message which contains its identity and geographical location. By the signal strength of the message, a neighbor of A^* can then determine its physical distance to A^* . Each neighbor then broadcasts a message indicating that its distance from A^* . A node that receives this message will determine its distance to A^* by adding its distance to its neighbor and the distance of its neighbor to A^* , which is carried in the message. Subsequently, every node in the network can identify the distance to A^* . *DV-hop* works in a similar manner except the physical distance is not derived from signal strength measurement but from *average distance per hop between anchor nodes*.

Theoretically, a normal node requires only distance and position information of three anchor nodes to perform triangulation. When there are more than three anchor nodes, a normal node can use the information of all the nodes to calculate its location or it can select only three. APS does not study this issue in detail. However, intuitively, different choices should yield different estimations of positions. An anchor node which is isolated in a remote area is not a good one since the distance estimated using *DV-hop* or *DV-distance* is usually deviate a lot from the real physical distance. We study the effect of selecting different anchor nodes using simulations.

B. Our Improved APS with Adaptive Anchor Selection

Generally speaking, the selection of anchor nodes [2] should be important to the precision of position estimation in most localization algorithms, especially true for the APS or specifically *DV-distance* algorithm in the presence of measurement error in anisotropic networks. Figure 1 shows a convincing example of the estimated positions by both the original and modified APS, that will exhaustively search for the best 3 anchors among all possible combinations of 3 anchors, for a specific sensor node against its true position.

For clarity of presentation, the square is used to denote the true position of the individual node of interest. The “*” symbol represents the estimated position by our modified APS whereas the circle is the estimated position by the original APS. Obviously, the estimated node position by the original APS is far from its true position for which the estimated position by our modified APS is much closer in this particular case. From our empirical experience, we find that similar cases occur for the

‘corner’ or its neighboring nodes around the two ends of the C-shape. In corners, the estimation error in the accumulated distance along the various paths from *all anchor nodes* as computed in the original *DV-distance* algorithm of the APS approach can be much more profound and misleading when compared to that of other positions in the C-shape, thus introducing a much higher imprecision in the estimated position as illustrated in Figure 1.

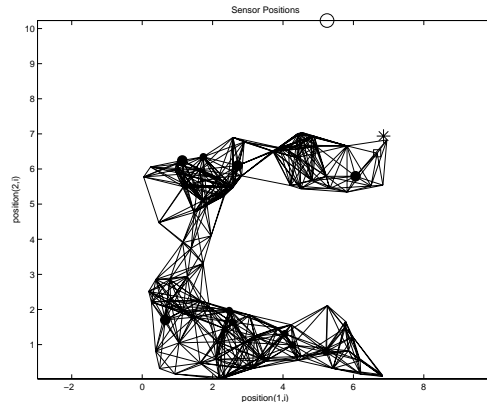


Figure 1. An example showing the benefit of selecting the best 3 anchors rather than all anchors in APS for a particular node

In view of the empirical results obtained for the modified APS, we proposed in our previous work [2] an *improved APS algorithm*, namely the *APS(Near-3)*, based on a simple yet effective heuristic as follows: the improved APS algorithm will always choose the *nearest 3* anchors for a faster response time with respect to each individual sensor node inside the original APS computation (i.e. the triangulation mechanism) used for its position estimation. Clearly, rather than choosing the nearest 3 anchors, our adaptive proposal allows us to carefully choose the k , where $k \geq 3$, most suitable anchors depending on the actual topology, connectivity information and other physical or environmental factors considered. To compare their performance, we implemented both the original and improved APS in the earlier Matlab Version 7.0.1 running on a Pentium IV desktop PC installed with 3 GHz processor. Among 20 test cases of anisotropic topologies, the best resulting topology obtained by the improved APS gives $0.1527R$, where R is the length of the unit square for placing the sensor nodes in simulation, as the average error whereas the worst result shows $0.5716R$, producing an overall averaged error of $0.3343R$ for all the 20 test cases. On the other hand, the original APS produces $0.7683R$ as the average error for the best case, with the worst result at $1.8731R$, and giving an overall averaged error of $1.0725R$ for all the 20 cases.

III. OUR ADAPTED MICRO-GENETIC ALGORITHMS

In this section, we firstly detail the original micro-genetic algorithm (MGA) [4] using two descent-based genetic

operators and a gene-pivoting scheme to solve general optimization problems. Next, we will consider how our original proposal of MGA can be adapted as a post-optimizer to further enhance the position estimation of our improved APS.

A. The Original Micro-Genetic Algorithm

Evolutionary algorithms (EA) [4, 19] are local search methods capable of efficiently solving complex constrained or unconstrained optimization problems (OPT) [1]. As previously discussed, localization in wireless sensor networks is intrinsically an unconstrained OPT. However, in handling relatively sizable (with ≥ 500 variables) or complicated OPTs, the total computational costs can be drastically reduced by focusing the search only on a reasonably small population of chromosomes without much impact on the search efficiency. In fact, it has been reported that MGAs can find solutions with fewer iterations than evolutionary algorithms (EAs) with larger population sizes for some problems [4]. Therefore, in this paper, we consider a micro-genetic algorithm (MGA) based on a small population size (usually < 30) to tackle the challenging localization problems in wireless sensor networks. Basically, a variable in an OPT is usually represented by a gene in a MGA. A chromosome, consisting of all the genes, is used to denote a valuation for all the variables. As inspired by the nature, an EA or MGA basically performs a parallel local search in different parts of the search space through maintaining a population of chromosomes for iterative improvements over the successive generations.

Figure 2 shows the pseudo-code of the original MGA proposed by Stuckey and Tam [4] for solving general constrained or unconstrained optimization problems.

```

MGA( $PZ, MZ, fitness()$ )
initialize a small Population of  $PZ$  chromosomes
repeat
  select the best  $MZ$  chromosomes  $\in Population$ 
   $Population := \emptyset$ 
  repeat
    produce  $\{offspring\}$  by descend-based genetic opr.
     $Population := Population \cup \{offspring\}$ 
  until ( $sizeof(Population) = PZ$ )
until ( $Population$  is converged or the resource limit is
  exceeded)

```

Figure 2. The Convergence Procedure of the Original MGA

Given the population size PZ , the size MZ of the mating pool and the evaluation function $fitness()$, the original MGA initially sets up a small population. For unconstrained optimization, the $fitness()$ function corresponds directly to the objective functions being considered in the original problem formulations. For constrained optimization, the number of constraint violations produced by each individual chromosome can be reflected in an augmented form of the original objective function to give the resulting $fitness()$ value with relatively larger penalties attributed to the constraint

violation part for a minimization problem. When there is no constraint violation, the constraint violation part in the augmented $fitness()$ function returns 0, thus guiding the search to work on the original objective function. In each generation, the original MGA selects the best MZ chromosomes according to $fitness()$ from the current *Population* to construct the mating pool in which specific genetic operators such as mutation or crossover can be applied to produce offspring to form the next generation. This “selection-and-reproduction” process of MGA is repeated until all the chromosomes have converged to the same local minima, or some predetermined resource limit is exceeded. A resource limit is usually defined in terms of CPU time or the maximum number of generations allowed.

To solve structured problems such as the constraint satisfaction problems (CSPs) [20] or constrained optimization problems [9], the original MGA employs two principal mutation operators, namely the min-conflict heuristic-based [19] *apply_MCH* operator and the opportunistic *single_pt_mutate* operator. Given a chromosome cs containing a gene V , the *apply_MCH* operator will exhaustively look for all possible values of the gene V to update the current valuation denoted by cs . The mutant cs' producing the smallest $fitness()$ value will be selected as the resulting chromosome. Ties will be broken randomly whenever they occur. In other words, the *apply_MCH* operator is the same as the variable updating mechanism as used in the GENET model [20]. Since the *apply_MCH* operator can be computationally expensive for a practical CSP involving hundreds of genes/variables, the original MGA requires a gene pivoting scheme to determine the “most important” gene of each chromosome on which the *apply_MCH* operator will be invoked. For the remaining genes in the chromosome, they will be updated by the *single_pt_mutate* operator which randomly selects a value for each gene under consideration. In case the mutant cs' achieves a lower $fitness()$ value when compared to that of the original chromosome cs , the mutant cs' will be chosen. Otherwise, the original cs is kept. It is worth noting that due to its highly diversifying search property, the crossover operator is not considered in the original MGA in solving structured problems like the CSPs. Nevertheless, as reported in [4, 19], the original MGA has produced remarkable results in solving a number of challenging CSPs or constrained OPTs including a set of hard graph-coloring problems [20] and complex job-shop scheduling problems [19].

B. Our Adapted Micro-Genetic Algorithm (MGA^*_{LOC})

The original MGA is targeted to solve hard CSPs or constrained OPTs involving variables of finite domains. Though with sufficient domain knowledge and carefully designed domain revision techniques, the original MGA should be easily adapted to solve continuous CSPs or constrained OPTs. However, according to we defined in Equation (1) in Section II, localization intrinsically as an unconstrained OPT involving continuous variables for position estimation does not have any problem constraint

to provide sufficient guiding information such as the node or arc consistency [20] to “initialize” or “prune” the search. Obviously, it can be difficult to directly use our adapted MGA to independently solve this *generalized* localization problem in wireless sensor networks. Therefore, we propose in this paper to adapt the original MGA as a “post-optimizer” to be *flexibly* integrated into various localization methods [3, 10, 12, 13, 16] so as to further reduce the overall estimation error of the best estimates returned by the underlying methods. As the first attempt to apply an evolutionary approach [4] to tackle localization in wireless sensor networks, this new proposal has a unique property of being adaptive to any existing localization method. Since we never make any restrictive assumption, like the existence of sufficiently close neighbors in the convex programming approach [3], about the physical environment of the underlying sensor networks in the generalized formulation of localization problems as given in Equation (1) of Section II, on which our adapted MGA are based, this simply implies that our adapted MGA can be integrative to any existing localization method. In a nutshell, the specific assumptions about the background or physical environment of the underlying sensor networks are being made in the relevant localization method but not in our adapted MGA, thus giving much flexibility to our adapted evolutionary approach.

Figure 3 shows the pseudo-code of our adapted MGA for localization, namely the MGA^*_{LOC} , to enhance the position estimation attained by any localization method.

```

MGA*LOC(Best-Values, PZ, fitness())
construct an initial Population of PZ chromosomes by
  flipping some values in Best-Values;
repeat
  foreach chromosome cs ∈ Population
    produce offspring1 by the descend-based mutation
    if (fitness(cs) > fitness(offspring1))
      replace cs with offspring1
    produce offspring2 by the descend-based crossover
    if (fitness(cs) > fitness(offspring2))
      replace cs with offspring2
  endforeach
until (Population is converged or the resource limit is
  exceeded)

```

Figure 3. The Convergence Procedure of our Adapted MGA

The *Best-Values* is the set of the current best position estimates returned by any existing localization method. The *PZ* is the population size of our adapted MGA whereas the *fitness()* function denotes the evaluation function for each individual chromosome depending on the underlying localization method used. Usually, the *fitness()* function we used is the same as the objective function used in the localization method for ease of comparison to demonstrate the ultimate improvement attained. For instance, for our improved APS method, the *fitness()* function can be formally defined as follows.

$$fitness(x', y') = \sum_{i=1}^3 \left(\sqrt{(x' - x_i)^2 + (y' - y_i)^2} - pdist_i \right) \quad (2)$$

where (x', y') represents the current position estimate, (x_i, y_i) is the absolute position of the anchor node i with $i = 1, 2, 3$ for the 3 nearest anchors as used in the original or improved APS method [2], and $pdist_i$ is the relative path distance measure of anchor node i with respect to the current position estimate. Clearly, the *fitness()* function gives the mean square error of the current position estimate (x', y') with respect to the positions of the 3 nearest anchor nodes and their relative path distance measures.

Basically, our adapted MGA initially constructs a small population of chromosomes based on flipping the currently best values of position estimation in the set *Best-Values* returned by any existing localization method. In each iteration, a chromosome *cs* taken from the current population will be systematically experimented with the two genetic operators, namely the descend-based mutation and crossover operators, aimed to produce a better *offspring* in term of its *fitness()* value. When a better *offspring* is produced, the original *cs* will be replaced by the *offspring*. The iteration continues until the topmost k (say 3) chromosomes are converged to the same solution, or the resource limit such as the number of iterations is exceeded. It should be noted that unlike the conventional EA, our adapted MGA does not employ any selection mechanism based on certain probability distribution like the Roulette Wheel mechanism [4] to bias toward those favorable chromosomes in the current population. For simplicity and better efficiency, we allow only the best *PZ* chromosomes remained in the current population after being ranked by the *fitness()* function, which will then be opportunistically improved by the two descend-based genetic operators over successive generations.

The two descend-based genetic operators for our adapted MGA are described as below.

- Descend-based mutation operator: for each chromosome in the current population, the operator will invoke the *rand()* function in C returning a probability p ranging from $0 \dots 1$. If $p > 0.5$, the mutation operator will generate a random point (x_r, y_r) in a straight line formed by the current position estimate (x', y') and the previous position estimate (x_{old}, y_{old}) . In case the newly generated mutate point (x_r, y_r) produces a descend in the objective value returned by *fitness()* when compared to that of the current estimate (x', y') , (x_r, y_r) will replace the current position estimate (x', y') as the new chromosome; otherwise, the current chromosome will remain unchanged. Essentially, the descend-based mutation operator will look for opportunistic improvement in each position estimate as denoted by each chromosome in the small population of our adapted MGA over successive generations.

➤ Descend-based crossover operator: basically, the whole population of chromosomes is readily sorted in descending order according to their objective values. The crossover operator looks for any opportunistic improvement between any pair of chromosomes sequentially extracted from the sorted list of chromosomes. In this regard, we usually set the PZ to be even. In case not, the last pair can simply be formed from the last and first chromosomes of the sorted list. For each pair $(i, i + 1)$ of chromosomes, when the probability value p as returned by $rand()$ is greater than 0.5 , the crossover operator will proceed to consider the mid-point as the crossover point of their position estimates. Similar to the mutation operator, in case there is any decrease in the objective value of this crossover point when compared to that of the worse chromosome $(i+1)$ in the pair, the chromosome $(i + 1)$ will be replaced by the new crossover point; otherwise, nothing occurs. The crossover operator will also be applied to the whole population over successive generations of our adapted MGA.

It is interesting to note that crossover operator is not used in the original MGA. However, in our adapted MGA for localization, we obtained from our empirical observation the descend-based crossover operator did sometimes help to diverge the search to a better solution. This is probably due to the huge search space of the underlying localization problems as “continuous” OPTs for which the diversifying nature of the crossover operator can be advantageous.

IV. EMPIRICAL EVALUATION

In this section, we evaluate the performance of our proposal of using adapted MGA as a post-optimizer to further enhance the *improved APS* method [2] utilizing the nearest 3 anchor nodes for localization in wireless sensor networks. To demonstrate the effectiveness of our adapted MGA, we randomly generated 5 test cases for various parameter settings for both uniform and C-shape topologies of wireless sensor networks to compare the performance of the *improved APS* method and the *improved APS* integrated with our adapted MGA as the *improved APS-MGA* method for localization. Both the *improved APS* and *improved APS-MGA* methods are based on the *DV-distance* algorithm [8] as described previously. For all test cases, the ratio of anchors is 10% of the total number N of sensor nodes. Besides, following [8], we assume a uniform distribution of 10% measurement errors across both uniform and anisotropic sensor networks to test on the robustness of the localization algorithms for comparison. For the subsequent discussion, the performance of each individual localization method is evaluated on the average estimation error in terms of R for the involved radio range. The improved APS algorithm is written as a Matlab program executable in the Matlab Version 7.2 system whereas our adapted MGA is implemented as

external C functions compiled using the Matlab *mex* compiler for invocation by the relevant Matlab functions. Their simulation results are obtained on a Pentium IV desktop PC installed with a 3 GHz processor and 512 megabytes of RAM under the MS Windows XP operating system. Besides, it should be noted that we arbitrarily set the convergence condition of our adapted MGA as whenever the best 3 chromosomes are converged to the same solution, the whole population is regarded as “converged” for quick experimentation. In addition, the maximum number of generations for each invocation of our adapted MGA to improve on the current position estimate is set to 1,000.

Table I compares the performance of the *improved APS-MGA* method integrated with our adapted MGA against our originally *improved APS* method in terms of the mean position estimation error on uniform sensor networks with the total number N of sensor nodes as 100 or 150, and the physical dimension R of the square area for spreading the sensors with $R = 1.2r, 1.6r, 2.0r$ where r denotes the predetermined communication range of each

TABLE I
THE PERFORMANCE OF THE IMPROVED APS-MGA AGAINST THE IMPROVED APS ON UNIFORM SENSOR NETWORKS

$R = 1.2r$	(100 nodes, 10% anchors)					
	1	2	3	4	5	Avg.
Impr. APS	0.207	0.655	0.187	0.256	0.397	0.341
Impr. APS-MGA	0.187	0.608	0.174	0.235	0.369	0.314
% of Impr.	9.9	7.2	7.6	8.3	7.1	8.0
$R = 1.6r$	(100 nodes, 10% anchors)					
Impr. APS	0.081	0.370	0.045	0.160	0.270	0.185
Impr. APS-MGA	0.079	0.334	0.038	0.152	0.253	0.171
% of Impr.	3.3	9.6	16.4	4.8	6.4	8.1
$R = 2.0r$	(100 nodes, 10% anchors)					
Impr. APS	0.053	0.281	0.016	0.081	0.174	0.121
Impr. APS-MGA	0.037	0.263	0.011	0.080	0.166	0.111
% of Impr.	31.2	6.5	32.9	1.7	5.0	15.5
$R = 1.2r$	(150 nodes, 10% anchors)					
Impr. APS	0.642	0.468	0.372	0.260	0.496	0.448
Impr. APS-MGA	0.385	0.462	0.313	0.200	0.452	0.363
% of Impr.	40.0	1.3	15.7	23.3	8.9	17.8
$R = 1.6r$	(150 nodes, 10% anchors)					
Impr. APS	0.382	0.328	0.211	0.105	0.328	0.271
Impr. APS-MGA	0.201	0.324	0.179	0.093	0.292	0.218
% of Impr.	47.3	1.3	15.1	11.6	10.9	17.3
$R = 2.0r$	(150 nodes, 10% anchors)					
Impr. APS	0.244	0.238	0.129	0.071	0.240	0.184
Impr. APS-MGA	0.141	0.237	0.118	0.060	0.225	0.156
% of Impr.	42.2	0.4	8.6	15.9	6.0	14.6

sensor node. Since our adapted MGA is used as a post-optimizer, with the initial, but sometimes possibly fairly rough, position estimate provided by the Matlab nonlinear optimizer, our adapted can always successfully improve on the original position estimates obtained by the improved APS method. To the worst, our adapted MGA may simply return the same position estimate in case it fails to search for a better solution, which has *never* occurred in our simulation tests. Therefore, we can consistently see improvements, no matter minor or major, in the mean position error as obtained from our adapted MGA when compared to that of the originally improved APS in Table I.

For better presentation and thus understanding of our comparison, we highlight only the lowest and highest improvements as obtained by our adapted MGA in bold-faced and italic figures among the two major groups of simulation results with $N = 100$ and 150 respectively with 10% anchors. As clearly shown in Table I, for $N = 100$, the highest improvement by our adapted MGA is 32.9% whereas the lowest improvement is 1.7%. For $N = 150$, the highest improvement by our adapted MGA is 47.3% whereas the lowest is 0.4%. For all the test cases, the highest improvement is 47.3% whereas the lowest is 0.4%, giving an overall average of 13.5% for our adapted MGA as a clear indicator to demonstrate the effectiveness of our proposal to the improved APS and possibly many other localization methods. Interestingly, for the uniform topologies with $N = 100$, the averaged % of improvement is increasing as 8.0%, 8.1%, 15.5% respectively for $R = 1.2r, 1.6r, 2.0r$, giving a group averaged improvement for $N = 100$ as 10.5%. On the other hand, when $N = 150$, the averaged % of improvement is decreasing from 17.8% to 17.3% and lastly to 14.6% for $R = 1.2r, 1.6r, 2.0r$ respectively, giving its group averaged improvement as 16.6%.

Figure 4 shows such interesting and opposite trends of changes in the averaged % of improvement by our adapted MGA on the uniform sensor networks with $N = 100$ and 150 respectively. In general, the mean ratio of connectivity increases from 10.75 to 26.08 for our test case with $N = 100$, and similarly from 10.87 to ultimately to 27.83 for uniform topologies with $N = 150$. Our conjecture for such phenomenon of opposite trends in the averaged % of improvement by our adapted MGA can be explained as follows. When the network size is relatively small, i.e. $N \leq 100$, with the relatively poor quality of the intermediate position estimates obtained from the improved APS method, there is still lot of room for improvement by the adapted MGA with the increasing sizes of the involved unit squares at $R = 1.2r, 1.6r$ and $2.0r$ respectively. Besides, the increasing ratio of connectivity between the nodes may provide useful proximity information to indirectly guide the “local” search of the adapted MGA towards better solutions. On the other hand, when the network sizes become larger, say $N \geq 150$, the relatively more diverse geographical distribution of the sensor nodes increases the absolute distance between the involved nodes as R increases. These increases in the absolute distance simply bring

greater uncertainty to the position estimation by our adapted MGA, and more importantly out-weighted the information gain due to the increased ration of connectivity for such medium- to large-sized sensor networks. Nevertheless, the specific reason(s) for such opposite trends of averaged % of improvement by our adapted MGA should prompt for further investigation. Undoubtedly, there is an increase in the group averaged % of improvement from 10.5% to 16.6% when N goes from 100 to 150. This clearly demonstrates the effectiveness of our adapted MGA in further enhancing the position estimates by the originally improved APS method. Apparently, the larger the network sizes, the higher the averaged % of improvement by our adapted MGA on the uniform sensor networks. After all, further experimentation and careful analysis would be required to investigate whether such increase in the group averaged % of improvement will *continue* for $N = 200$ or larger, and more importantly the detailed explanation accountable for such change(s) in the averaged improvement attained by our adapted MGA on uniform sensor networks.

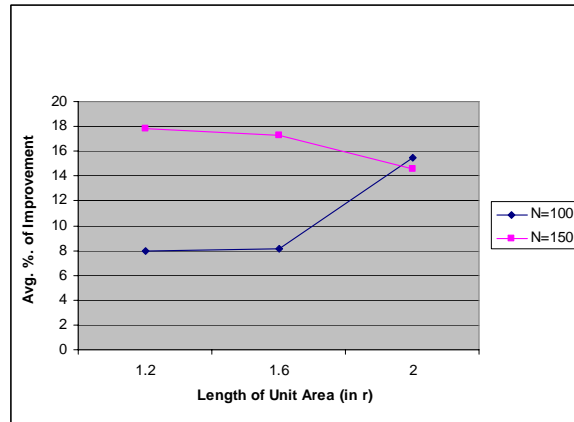


Figure 4. The Change in the Averaged % of Improvement for Our Adapted MGA on Uniform Sensor Networks.

Table II gives the performance of the improved *APS-MGA* method against the improved *APS* on the C-shape topologies of anisotropic sensor networks with $N = 100$ or 150 , and R ranging from $1.2r, 1.6r$ to $2.0r$ respectively. Similar to Table I, among the two major sets of results for $N = 100$ and 150 respectively, the lowest and highest % of improvements by our adapted MGA are highlighted as bold-faced and italic figures for clarify of presentation. When $N = 100$, the highest improvement by our adapted MGA is 24.7% whereas the lowest improvement is 0.5%. For $N = 150$, our adapted MGA produces its highest improvement at 66.3% and its lowest at 6.0%. In general, the highest improvement for all the test cases is 66.3% whereas the lowest is 0.5%, giving an overall average of 18.6% to reveal the “persistent” strength of our adapted MGA to enhance the improved APS and possibly many other localization methods on anisotropic sensor networks. Furthermore, for these C-shape topologies with $N = 100$, the averaged % of improvement is increasing as

8.0%, 8.3%, 8.8% respectively for $R = 1.2r$, $1.6r$, $2.0r$, giving a group averaged improvement as 8.4%. Contrary to the previous results obtained on uniform topologies, when $N = 150$, the averaged % of improvement is still increasing and even more obviously from 22.3% through 29.9% and ultimately to 34.3% for $R = 1.2r$, $1.6r$, $2.0r$ respectively, giving its group averaged improvement as 28.8%.

Figure 5 gives a plotting to clearly reveal these attractive increases in the averaged % of improvement by our adapted MGA on the anisotropic sensor networks with $N = 100$ and 150 respectively. This is probably because in the anisotropic topologies, the uneven distribution of sensor nodes and the unbalanced path distance measures in different directions simply introduce greater measurement and thus estimation errors in the some specific cases such as those nodes around the corners of the C-shape by the concerned localization methods including our improved APS. Such specific cases may open up wider opportunities for our adapted

TABLE II.
THE PERFORMANCE OF THE IMPROVED APS-MGA AGAINST THE IMPROVED APS ON ANISOTROPIC SENSOR NETWORKS

$R = 1.2r$	(100 nodes, 10% anchors)					
	1	2	3	4	5	Avg.
Impr. APS	0.922	0.826	0.833	0.358	0.319	0.652
Impr. APS-MGA	0.825	0.797	0.822	0.336	0.259	0.608
% of Impr.	10.4	3.5	1.3	6.2	18.7	8.0
$R = 1.6r$	(100 nodes, 10% anchors)					
Impr. APS	0.675	0.539	0.579	0.226	0.192	0.442
Impr. APS-MGA	0.604	0.516	0.570	0.225	0.144	0.412
% of Impr.	10.5	4.3	1.5	0.5	24.7	8.3
$R = 2.0r$	(100 nodes, 10% anchors)					
Impr. APS	0.531	0.429	0.396	0.169	0.151	0.335
Impr. APS-MGA	0.480	0.405	0.388	0.163	0.116	0.310
% of Impr.	9.6	5.6	2.1	3.5	23.2	8.8
$R = 1.2r$	(150 nodes, 10% anchors)					
Impr. APS	0.273	0.228	0.632	0.317	0.205	0.331
Impr. APS-MGA	0.204	0.159	0.594	0.276	0.129	0.272
% of Impr.	25.3	30.4	6.0	13.1	36.8	22.3
$R = 1.6r$	(150 nodes, 10% anchors)					
Impr. APS	0.150	0.152	0.452	0.234	0.087	0.215
Impr. APS-MGA	0.080	0.101	0.425	0.208	0.041	0.171
% of Impr.	46.3	33.4	6.2	11.2	52.3	29.9
$R = 2.0r$	(150 nodes, 10% anchors)					
Impr. APS	0.105	0.114	0.333	0.160	0.062	0.234
Impr. APS-MGA	0.035	0.076	0.308	0.141	0.030	0.118
% of Impr.	66.3	33.5	7.5	12.2	51.8	34.3

MGA to further enhance the immediate results obtained by our improved APS. More importantly, without making any explicit assumption about the local/global topologies, and the locations of the concerned sensor nodes or their neighbors, our adapted MGA would never be misled by such explicit assumption possibly failed in various individual nodes in anisotropic networks. This is evident from the fact that the averaged estimation error over 5 random test cases by the improved APS on anisotropic networks is always relatively higher than that on the uniform networks at the same parameter setting. For instance, the averaged estimation error over 5 test cases for the improved APS on anisotropic networks with $N = 100$ and $R = 1.2r$ is 0.652 whereas that by the improved APS on uniform sensor networks is 0.341 that is close to half of the result obtained for anisotropic networks. With the same averaged measurement error as 10% for both uniform and anisotropic topologies in our simulation tests, this may possibly imply that there are much greater room for improvement by our adapted MGA on anisotropic topologies. Besides, as indicated in Figure 5, the opportunities for greater improvement by our adapted MGA would generally increases when the network size grows, or when the physical dimension R of the unit square rises. After all, the actual change(s) exhibited by our adapted MGA for larger network size or physical dimension of anisotropic networks, and more importantly the detailed reason(s) behind such change(s) prompt for our further investigation.

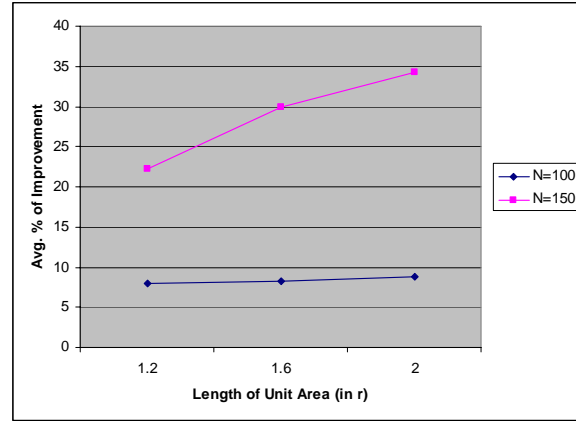


Figure 5. The Change in the Averaged % of Improvement for Our Adapted MGA on Anisotropic Sensor Networks.

Lastly, our adapted MGA is efficient for execution. The averaged CPU times as measured by the relevant timing function in the Matlab Version 7.2 system for executing our adapted MGA as external C functions on a Pentium IV PC installed with a 3 GHz processor, 512 megabytes of RAM and the Windows XP operating system are 9.69 seconds and 21.06 seconds respectively for $N = 100$ and 150 on uniform sensor networks. For anisotropic networks, when $N = 100$ and 150 , the averaged CPU times are only slightly increased to 10.32 seconds and 24.95 seconds respectively, again showing the efficiency of our proposal.

V. CONCLUDING REMARKS

Wireless sensor networks are widely applicable to many practical applications including environmental monitoring, military applications, disaster management, etc. in which the geographical locations of sensors are important. Accordingly, various localization algorithms have been developed with a few anchor nodes having their own positions precisely determined. Undoubtedly, most of the existing works focus on increasing the accuracy in position estimation by using a single heuristic-based or mathematical technique. An example is the Ad-hoc Positioning System (APS) [10] that works by the triangulation mechanism used in the Global Positioning System (GPS). Extending from our previous work [2] in which we carefully considered the impacts of anchor selection in solving the localization problems, we proposed in this paper to adapt a generic micro-genetic algorithm (MGA) [4, 19] for integration into various localization algorithms as a post-optimizer for further position refinements whenever possible. Mainly, our adapted MGA for localization relies on two genetic operators, namely the descend-based mutation and crossover operators, aiming at opportunistic decreases in the objective values obtained for the new mutate out of the current position estimate, or the crossover point for any pair of existing chromosomes over successive generations. Unlike the conventional evolutionary approach, our adapted MGA never employs any selection mechanism like the Roulette Wheel mechanism [4] to bias toward those favorable chromosomes in the current population. In each iteration, our adapted MGA allows only the better mutant generated out of the two genetic operators to replace its parent and then simply retain all the chromosomes in the population for other possible improvements until the best k (say 3) chromosomes are converged to the same solution, or the resource limit such as the number of iterations is exceeded.

To demonstrate the feasibility of our MGA proposal, we implement the *improved APS* algorithm in the Matlab Version 7.2 with our adapted MGA as external C functions interfacing to Matlab functions, and evaluate their simulation results on a total of 60 randomly generated test cases for both uniform and C shape topologies of wireless sensor networks. Our simulation results clearly demonstrate the effectiveness of our adapted MGA proposal as the post-optimizer in reducing the average estimation error maximally to around half (precisely 47.3% and 66.3%) of that attained by the originally improved APS method on both uniform and anisotropic sensor networks. The averaged percentages of improvement by our adapted MGA are 13.5% and 18.6% respectively for uniform and anisotropic sensor networks. More importantly, our adapted MGA, without any prior assumption/knowledge about the underlying localization method, is so generic that it can readily be integrated into any existing localization algorithms [2, 8, 12, 16] for further position refinement.

This work opens up a number of interesting directions that are worth exploring. First, as pointed out in Section IV, we should conduct more experiments and careful

analysis to explain in detail about the general increase in the percentage of improvement when the total number N of sensor nodes increases, and more importantly the reason(s) accountable for the specific change(s) across the different values of R at the same N . Besides, it should be interesting to study about the integration of our generic MGA proposal into other localization algorithms. Lastly, it is worth investigating on refining the two descend-based genetic operators, possibly by fine tuning of the involved parameters or integrating with more sophisticated selection criteria, so as to further improve the overall accuracy of position estimation by our adapted MGA.

ACKNOWLEDGMENT

The authors wish to thank Professor Yi Shang for his fruitful discussion.

REFERENCES

- [1] *Nonlinear Programming (2nd Edition)*. Athena Scientific, 1999.
- [2] K. Cheng, V. Tam, and K. Lui. Improving APS with anchor selection in anisotropic networks. In *Proceedings of the International Conference on Networking and Services (ICNS'05)*, October 2005.
- [3] L. Doherty, K. Pister, and L. Ghaoui. Convex position estimation in wireless sensor networks. In *IEEE Infocom*, 2001.
- [4] G. Dozier, J. Bowen, and D. Bahler. Solving small and large scale constraint satisfaction problems using a heuristic-based microgenetic algorithm. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1994.
- [5] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the physical world with pervasive networks. In *IEEE Pervasive Computing*, pages 59 – 69, 2002.
- [6] C. Gentile. Sensor Location through Linear Programming with Triangle Inequality Constraints. In *Proceedings of the IEEE International Conference on Communications (ICC 2006)*, Istanbul, Turkey, June 11-15, 2006.
- [7] X. Ji and H. Zha. Sensor positioning in wireless sensor networks using multidimensional scaling. In *IEEE Infocom*, 2004.
- [8] H. Lim and J. C. Hou. Localization for anisotropic sensor networks. In *IEEE Infocom*, 2005.
- [9] K. Marriott and P. J. Stuckey. *Programming with Constraints*. MIT Press, 1998.
- [10] D. Niculescu and B. Nath. Ad hoc positioning system (aps). In *IEEE Globecom*, pages 2926 – 2931, 2001.
- [11] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. In *Communications of the ACM*, volume 43, pages 51 – 58, 2000.
- [12] A. Savvides, H. Park, and M. Srivastava. The bits and flops of the nhop multilateration primitive for node localization problems. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.
- [13] Y. Shang and W. Ruml. Improved MDS-based localization. In *IEEE Infocom*, 2004.
- [14] Y. Shang, W. Ruml, and Y. Zhang. Localization from mere connectivity. In *ACM MobiHoc*, 2003.
- [15] SwarmINTEL.com. Swarm intelligence resources. In *website available at: <http://www.swarmintelligence.net>*, 2004.

- [16] V. Tam, K.Y. Cheng and K.S. Lui. Improving localization in Wireless Sensor Networks with An Evolutionary Algorithm. In *Proceedings of The IEEE Consumer Communications and Networking Conference (CCNC'06)*, Las Vegas, NV, USA, January 8 – 10, 2006.
- [17] V. Tam, K.Y. Cheng and K.S. Lui. Hybrid Approach for Localization in Anisotropic Sensor Networks. In *Proceedings of The IEEE 63rd Vehicular Technology Conference (VTC2006-Spring)*, Melbourne, May 7 – 10, 2006.
- [18] V. Tam, S. Koo, and K. Sugiyama. Evolving artificial ant systems to improve layouts of graphical objects. In *Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence (PRICAI'04)*, pages 955 – 956, 2004.
- [19] V. Tam and P. Stuckey. Improving evolutionary algorithms for efficient constraint satisfaction. In *The International Journal on Artificial Intelligence Tools*, volume 28, pages 363 – 383, 1999.
- [20] E. Tsang. *Foundations of Constraint Satisfaction*. Academic Press. 1993.

Vincent Tam obtained his Master of Philosophy in computer science from the Chinese University of Hong Kong in 1995, and completed his Ph.D. in computer science from the University of Melbourne, Australia in 1998.

He had previously worked as an academic fellow in the School of Computing (SoC), the National University of Singapore (NUS). Currently, he is a teaching consultant in the Department of Electrical and Electronic Engineering, The

University of Hong Kong. His current research interests include local search methods, localization and wireless sensor networks.

Dr. Tam is a member of the IEEE. He is listed in the Marquis's 9th Edition of Who's Who in Science and Engineering. Dr. Tam's papers in the IEEE ICTAI'94 and INTELLIGENCE & SYSTEMS'99 were selected as the best papers which were later invited for journal publications. Besides, Dr. Tam was the winner of Innovative Teaching Award among 80+ staff in the SoC, NUS in 2000.

King-Yip Cheng was born in Hong Kong. Mr. Cheng obtained his Bachelor of Engineering from the Department of Electrical and Electronic Engineering, the University of Hong Kong in 2004. He is currently studying the Master of Philosophy in the same department, and expecting to complete his study by the end of 2006. His current research interest mainly focuses on localization in wireless sensor networks.

King-Shan Lui obtained her BEng. (first class honors) and MPhil. degrees in computer science from the Hong Kong University of Science and Technology. She then received her PhD degree, also in computer science, from the University of Illinois at Urbana-Champaign, USA, in 2002.

She joined the Department of Electrical and Electronic Engineering, the University of Hong Kong, as an assistant professor in August 2002. Her research interests includes QoS issues, protocol and algorithm design in the Internet, ad hoc networks, and sensor networks.

Dr. Lui is a member of IEEE.