

SCHOOL OF INFORMATION TECHNOLOGY
& ELECTRICAL ENGINEERING
THE UNIVERSITY OF QUEENSLAND
Brisbane
Queensland 4072
Australia

Phone: +61 7 3365 1009
Fax: +61 7 3365 4999
Email: tricia@itee.uq.edu.au

Technical Report
No. 465

*On optimal route computation of mobile sink in a
wireless sensor network*

*S. Nesamony, M. K. Vairamuthu, M. E. Orłowska and
S. W. Sadiq*

11/12/2006

On optimal route computation of mobile sink in a wireless sensor network

S. Nesamony, M. K. Vairamuthu, M. E. Orłowska and S. W. Sadiq

School of ITEE, The University of Queensland, Australia

{sudar,madhan,maria,shazia}@itee.uq.edu.au

ABSTRACT

There is evidence of a range of sensor networks applications where a mobile sink entity (node) is utilised for data collection from statically positioned sensor nodes in a sensor field. The mobile sink is typically required to cover the sensor field by physical motion in order to obtain the values from the sensor nodes in a periodic fashion. This characteristic leads to a very interesting problem of determining the optimal route of the mobile sink, in terms of distance travelled, to accomplish the data collection from all the sensor nodes. This minimum distance problem that is spanned from the design nature of the network has very intriguing and motivating connections with a set of classic computational problems. These cohesions and similarities are explored in this paper, and the computational complexity is analysed. The applicability of numerical solutions to the current problem is discussed and a numerical heuristic is provided to arrive at an approximate answer that is ‘close’ to the actual solution. An evaluation of the proposed approach is also provided through experimental results.

Keywords: Sensor Networks, Sensor Network Design, Optimisation problem, Travelling Salesperson Problem with Neighbourhoods, TSPN, Numerical Method Heuristic

1. INTRODUCTION

Wireless sensor networks are becoming indispensable on many fronts including consumer applications, civil applications, warehousing applications and military applications because of several distinct advantages over traditional methods. The

perfect juxtaposition of automation, computation and sustenance along with the decreasing cost of deployment makes sensor networks a perfect ingredient in many data collection, monitoring and distribution applications.

In this paper, we are concerned with a specific potential sensor network application that is basically characterised as follows: A collection of n sensor nodes are positioned stationary in the sensor field. There is a base station or a sink node which collects data from all the positioned sensors. The action that the sink can take can be mere propagation of data to further systems or computation and initiation of some events based on the collected data or computed result.

Consider for example, a sensor field which is large with respect to the transmission range of the individual sensors and contains a heavy population of the sensors. An approach, to collect the data from these sensors which may have non-overlapping ranges, is proposed to be to use the sink as a mobile agent that physically moves around the sensor field and collecting the data from the sensors by visiting their transmission ranges.

The communication architecture of the network could be designed based on the requirements and limitations of the application. For example, the sensor nodes can be designed to transmit data if and only if requested by the sink node thereby reducing the energy consumption or wastage.

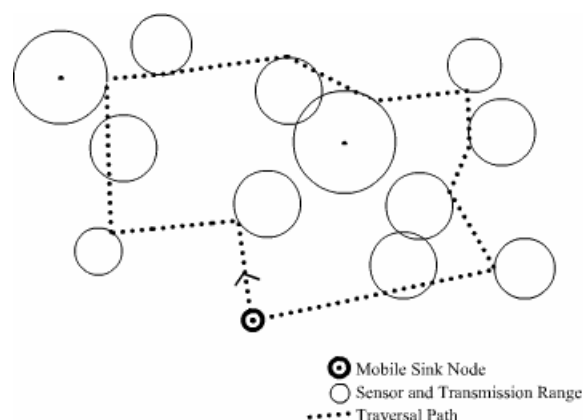


Figure 1. Sensor nodes with ranges and the mobile sink's path through the field

For collecting the data, the sink has to move around the field and position itself within the transmission range of every sensor node, query and collect the data thereby covering all the sensor nodes present in the field. The sought objective function in this scenario is to find such a route for the mobile sink to travel in the sensor field subject to the constraint that the distance traced by that route is minimum. A more formal representation of the problem is presented in the following sections.

The paper is organised as follows. Initially, the foundation concepts of the problem scenario and the domain it fits into are examined in section 2. The associated computation complexity is explored as well in the same section. Following the discussion about the complexity of the problem, is the section devoted to a comprehensive coverage of the related work that has been done towards the combinatorial problem to which our problem maps to. There is a brief discussion in section 4, about the approach taken towards solving the problem. It also contains the description of the sub problems that form the building blocks of the solution to the main problem. Section 5 is dedicated to the detailed discussion of the solution to the main problem. It is followed by the discussion of ordering of nodes to visit which plays a crucial role in the effectiveness and the efficiency of the final solution, in section 6. The retrospection and the analysis performed over the experimental results and the resultant observations are presented in the penultimate section 9. The summary of the overall paper along with the areas of further extension is provided in the concluding section.

2. FOUNDATION CONCEPTS

Evidently, the problems of shortest path or minimum distance have been considered from centuries ago in various forms and applications. The generic form of these problems is to determine a path or a route with minimum cost where the cost is modelled in different metric parameters depending on the application. In our case the metric considered for optimality is the distance travelled by the sink node.

Finding the minimum distance route taken by the mobile sink in order to visit the transmission range of all the sensor nodes, is to find a loop that touches a set of circles (assuming that the transmission area of all the sensor nodes are circular in shape) and a given point (the starting point of the sink node) whose length is minimum. Thus there is a perfect geometric inclination from the given real world problem. We operate on two dimensional Euclidean geometric which is explained further in this section.

With circles involved in estimation of the minimum distance loop, we consider the representation of circles to be a set of discrete points rather than values from the intervals of the real line. This discretisation is the first approximation step towards the sought best possible solution. The discretisation of the representation of circles will result in a small error factor of the solution reached depending on the frequency of discretisation of the circular segments as well as the rounding off errors.

The problem at hand is a combinatorial optimisation problem which can be formulated in terms of discrete optimisation where the variables present in the objective function are allowed to assume discrete values, namely the x, y coordinates identifying the point on the circumference of a circle in the two dimensional space.

The input to geometric minimum distance algorithms is a set of points in d -dimensional real space \mathbf{R}^d given by their coordinates. For $p \geq 1$, the distance between two points $(x_1, \dots, x_d), (y_1, \dots, y_d) \in \mathbf{R}^d$ in the ℓ_p norm is defined as $\sqrt[p]{\sum_{i=1}^d |x_i - y_i|^p}$ [1].

When $p = 2$, the norm will be the Euclidean norm which is used in this paper. The distance metric for our problem is measured in Euclidean geometric in two dimensions.

Conceptually the optimisation problem to be addressed is detailed as follows: Given a set of points $P \in \mathbf{R}^2$ in the Euclidean plane, and $n+1$ subsets $\{S_0, S_1, S_2, \dots, S_n\}$ of P where S_0 represents a fixed point and S_1 through S_n contains the points that are on the circumference of the circles thereby representing connected geometric regions in the Euclidean plane.

S_0 represents the fixed starting position of the mobile sink and $\{S_i\}_{i=1}^n$ represent the points that enclose the transmission region of the sensor nodes. The set S_0 is a singleton containing only one point and other sets contain points that lie on the circumference of the circles. Thus, $S_0 = \{a_{01}\}$, $S_1 = \{a_{11}, a_{12}, \dots, a_{1|S_1|}\}$, \dots , $S_n = \{a_{n1}, a_{n2}, \dots, a_{n|S_n|}\}$ and in general, $S_i = \{a_{ij}\}_{j=1}^{|S_i|}$.

The problem at hand is to construct a route on a set $P' \subseteq P$ such that P' contains at least one point from each subset S_i . The objective is to minimize the length of such a route.

Given a set of points $P \in \mathbb{R}^2$ in the convex Euclidean region, and $n+1$ subsets $\{S_0, S_1, S_2, \dots, S_n\}$ of P , the objective is to find

$$\min d_{\pi\varphi} \dots \dots \dots (1)$$

where

$$d_{\pi\varphi} = \sum_{i=0}^n d_{a_{\pi(i)} a_{\varphi(\pi(i+1) \bmod (n+1))}} \dots \dots \dots (2)$$

such that

- $d_{a_{ij} a_{kl}}$ being the Euclidean distance between the points (x_{ij}, y_{ij}) and (x_{kl}, y_{kl}) given by a_{ij} and a_{kl}

$$d_{a_{ij} a_{kl}} = \sqrt{(x_{ij} - x_{kl})^2 + (y_{ij} - y_{kl})^2} \dots \dots \dots (3)$$

- π is the permutation over n
 $\pi: \{0, \dots, n\} \rightarrow \{0, \dots, n\} \dots \dots \dots (4)$

$\pi(i)$ denotes the point which is at i^{th} location in the tour

- φ is the permutation over $\pi(i)$
 $\varphi: \{1, 2, \dots, |S_{\pi(i)}|\} \rightarrow \{1, 2, \dots, |S_{\pi(i)}|\} \dots \dots \dots (5)$

In the objective function (2) which is to be minimised, there is a permutation over another permutation thus contributing to the explosively large solution space where

the exploration of the solution occurs. Without any loss of generality, if we can assume that all the sets S_1 through S_n are of same cardinality of size M , that is, all the circles are characterised by M points that lie on the circumferences. Then, the brute force search of all the minimum length tours covering all the sets will be of the order of $n!M^n/2$ (halved because of the cyclic permutation of ordered tours) which is exorbitantly large.

The computational complexity of this combinatorial optimisation problem can also be determined by considering it a generalisation of the well-known Travelling Salesperson Problem (TSP). When the radii of all the circles (transmission ranges of all the sensors in the field), becomes zero or very close to zero, then this problem reduces to the TSP for $n+1$ points on an Euclidean plane which is NP-hard [2][3]. Hence the problem at hand is at least NP-Hard, because each circle can be a point at the simplest possible case. When the circles are of non-zero radii, this complexity will only escalate owing to all the points on the circumference of the circles. There is an exact isomorph of our problem which is a more generalised variant of TSP which is termed as TSPN (Travelling Salesperson Problem with Neighbourhood). A discussion about TSP and TSPN is presented in the following section.

3. RELATED WORK

Although the origins of the TSP can be traced back to 19th century, it is the paper [4] which came in 1954 that brought it into limelight. Since then this problem's popularity has soared high and it has been a classic problem in the domain of combinatorial optimisation.

General heuristic mechanisms that are devised for combinatorial optimisation problems are used to approach a general TSP problem. Also, given that it is a very hard problem to solve, several approximation algorithms have been proposed and the common line of interest is to find the approximation algorithm that finds a TSP tour which is at most c times the optimal one, where c is a positive constant. The popular Christofides algorithm [5] gave an approximation algorithm which gives a tour not

more than 1.5 times the optimal tour. The TSP for this algorithm considers the distance function to be symmetric and constrained to satisfy the triangle inequality. In fact this algorithm was one of the first approximation algorithms and cemented the position of approximation algorithms as a practical approach to intractable problems. The approximation algorithms generally provide provable solution quality and provable run time bounds as against heuristics which provide reasonable good solutions in reasonable time. Considering the PTAS (Polynomial Time Approximation Scheme) for the TSP, it is proved that it is NP-Hard to find a TSP tour that is at most $\frac{1}{219}$ more than the optimal tour [6]. With bounded metrics, it is also proved that there is no polynomial algorithm that finds a tour that is not more than $\frac{389}{388}$ times the optimal tour [7] unless $P = NP$.

Considering the extensive attention paid to the TSP and related problems, it is natural to observe that most of the different forms, flavours and variations of the classic TSP problem have been identified and attempted for deriving solutions. There are various forms of the problem like Max TSP, Min Area TSP, Max Area TSP, TSP with Neighbourhoods, Lawn Mowing Problem, Milling Problem etc. A brief discussion of these problems and other related network optimisation problems is presented in [8]. These problems that revolve around the actual core TSP have been studied exhaustively. So is our problem, which is in fact a specific version of the problem that is referred to as Travelling Salesperson Problem with Neighbourhood (TSPN). Although TSP has been under significant attention from the research community for more than half a century, the geometric TSPN has actually been formally addressed and treated only since 1994 in the paper [9] where the authors introduce it as an extended geometric version of the covering salesman problem [10].

The TSPN problem is defined over the scenario where the salesperson wants to meet a set of potential buyers each of who specifies a compact set in the plane, called his/her neighbourhood, within which he/she is willing to meet. The salesperson wants to compute a tour of shortest length that intersects all of the buyers' neighbourhoods and finally returns to his initial departure point. In the same paper [9], there are heuristic procedures for neighbourhood types such as parallel unit segments, translate of convex regions e.g. unit circle or rectangle, etc. The neighbourhoods are represented

by single points over which approximation algorithms are applied. The authors also present a Combination Lemma which approximates the problem with regions of different types, by combining approximations of each type. So, [9] gives a constant approximation ratio algorithm where the neighbourhoods are well behaved in specific forms. There is an $O(n)$ time $\sqrt{2}$ approximation algorithm for the case where the neighbourhoods are straight lines in the plane given by [11].

For the general case of connected arbitrary polygonal neighbourhoods, [12] gave an $O(\log k)$ approximation algorithm that is based on guillotine rectangular subdivisions [13] for TSPN with time complexity $\Omega(n^5)$ in the worst case, where k is the number of neighbourhoods (polygon regions) and n is the total complexity of the input. [14] came up with a significantly improved logarithmic approximation algorithm for the general case with running time $O(n^2 \log n)$. To be specific, the paper offers several results. The authors provide an algorithm that generates a tour with logarithmic approximation factor when the start point is known. If there is no start point given, then it is shown how to compute a good start point in $O(n^2 \log n)$. In addition, the authors provide an algorithm that performs at least one of the tasks that are listed below: (i) It outputs a TSPN tour of length $O(\log k)$ times the optimum tour in $O(n \log n)$ time (ii) It outputs a TSPN tour of length that is at most $(1 + \varepsilon)$ times the optimum tour in time $O(n^3)$ if $\varepsilon \leq 3$ or $O(n^2 \log n)$ otherwise, for any fixed $\varepsilon > 0$ arbitrary real constant as an optional parameter. It is not known in advance which of the above tasks will be accomplished as it depends on the instance of the neighbourhoods.

However, no polynomial time method guaranteeing a constant factor approximation is known for general connected regions as neighbourhoods. In general, if the neighbourhoods have similar size and shape, then one can usually find a constant factor approximation algorithm. If the neighbourhoods are of arbitrary size, then there could be a logarithmic approximation algorithm. Nevertheless, in [15], the authors give an $O(1)$ approximation algorithm, where the neighbourhoods are connected, disjoint, convex and fat. This was the first algorithm that did not require the neighbourhoods to have roughly the same size. There is a general graph version of the

TSPN problem which is called One-of-a-Set TSP [16], where the neighbourhoods may be disconnected. The direct use of the TSPN problem has been seen in applications like communication network design [17], VLSI routing [18] etc.

It is very interesting to note the notoriety of the TSPN problem and the opaque resistance it exhibits to the attempt at solutions. It is very hard to approach and solve the problem in general. The works that have been discussed so far concentrate on characterising the input i.e. identify the different styles of instances of the TSPN problem, and provide smart algorithms having theoretical bounds. Various characteristic setups of the problem instance are considered and the behaviour of execution of finding the TSPN tour is analysed in each case. It was shown that TSPN problem is in fact APX-hard when the input neighbourhoods are very long, skinny and overlapping [15][19] and cannot be approximated within a factor of $\frac{391}{390}$ unless P=NP. i.e. it is NP-Hard to find a $\frac{391}{390}$ approximation to TSPN. The APX hardness heavily relies on different sizes and overlap. If an optimisation problem belongs to the class of APX-hard, it is very hard problem to approach as it denies the existence of a PTAS and very difficult even to approximate. This APX-hardness proof was derived from the reduction of the known APX-hard problem, Min Vertex Cover problem.

With this appreciation of the deep complexities of the TSPN problem, let us proceed to analyse the problem that was identified in a specific design requirement of a sensor network and try to propose a numerical solution to it by reduction of search space and improved iterative approximation.

4. BASIC BLOCKS OF MINIMUM DISTANCE PROBLEMS

In this paper, we attempt to describe a practical problem, which emerges during the design phase of a specific wireless sensor network application scenario. In this network setting, the sensor nodes deployed in the field are positioned at fixed locations and the sink node that collects data from the sensors is mobile. The problem is to find the optimal route of the sink node in terms of the distance travelled such that the sink returns to its starting position and must have collected data from all the

sensor nodes in the field by positioning itself within the transmission range of every node. This real problem in the sensor network domain is identified to be an instance of the TSPN problem. In the previous section, the papers that address the hardness of the TSPN problem and give approximation algorithms for specific instances have been discussed.

This paper provides a geometry based heuristic that uses iterative procedures over the numerical method approach for the problem at hand. The evaluation of the performances is supported by experimentations. The heuristic is developed based on few components or sub-procedures. They provide a conceptual reasoning and understanding of the orientation of the approach. Also they lay the foundation for the final solution as well. These individual elements are described in detail in the following section. Some of these sub problems are hierarchically linked so as to offer the final solution which is ought to be very reasonable both in terms of the running time as well as accuracy.

Thus, in order to achieve the final objective of computing the minimal distance route in the given setup, we consider a set of minimum distance sub problems of increasing conceptual and computational complexity. They are individually discussed in this section and solutions are provided which are either direct or heuristic that aims at the best possible solution. These problems will provide the basic blocks of reasoning and understanding that will contribute to the final solution of our problem. We follow a bottom up approach of defining and solving individual problems and then using them as building blocks or components to compose the eventual solution.

These sub problems are all minimum distance problems containing points, lines and circles. There are three entities in every problem which are either points and lines or points and circles. There are four sub problems discussed viz. (i) Two Points and One Line (ii) Two Points and One Circle (iii) Two Lines and One Point and (iv) Two Circles and One Point. In line with these names our problem in fact can be termed as n-Circles and One Point which is discussed in the next section.

4.1 Two Points and One Line

The fundamental basic block of the whole problem of the minimum path distance is expressed in this sub problem. Let us consider a straight line and two points in a two dimensional Euclidean space. With respect to the notations mentioned earlier, we have $S = \{S_0, S_1, S_2\}$ where S_0 and S_1 are the single element sets containing a point each and S_2 is the set of points that denote a straight line. The objective is to find a point on the line S_2 such that the sum of distances between the points to the point on the line is minimum. It is a note here that since we are considering a whole line rather than just a segment, the set S_2 contains infinite elements, but this not of concern as the solution point in S_2 will be in between the (perpendicular) projections of S_0 and S_1 on S_2 . The solution is trivial if the points were on the opposite side of the line, it being the intersection point between the line S_0 and S_1 , and the given line S_2 . If the points S_0 and S_1 are on the same side of S_2 , then there is a direct method to solve this problem in real space, given by Heron of Alexandria [20] in his work Catoprtica.

The procedure given by Heron is very simple and direct that gives the exact solution. In accordance with our earlier notations, the elements of the sets S_0 and S_1 are given by, $S_0 = \{a_{01}\}$ and $S_1 = \{a_{11}\}$. Considering S_2 as a mirror, plot another point a'_{11} on the other side of the line S_2 that will be a mirror image of the point a_{11} . Thus, S_2 will be a perpendicular bisector of the line segment that connects the points a_{11} and a'_{11} . Now, the point of intersection of the line connecting a'_{11} and a_{01} and S_2 , which we may call a_{2x} is the required point on S_2 . This proof for this procedure is very simple and is based on the triangle inequality.

The solution given by Heron also follows the Principle of least action in Optics [21], and a consequence is that, the given two points form lines to the optimum point on the line, which sweep equal angles to the normal to the given line drawn at the optimum point.

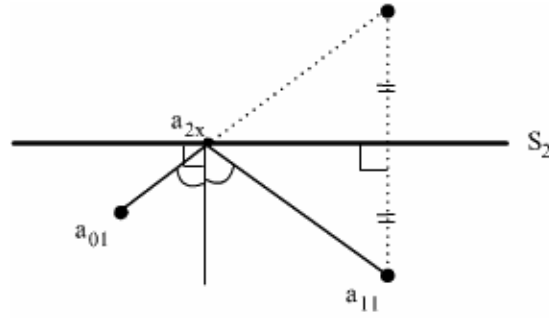


Figure 2. Equal angles to the normal at the optimum point

Although this least distance problem has no direct relevance with our problem, it is presented here as an acknowledgement to the simplest version of minimum distance loop problem. Also, this problem helps in supporting the future arguments in later sub sections with the concept behind equal angle sweeping. It also demonstrates the underlying semantics of minimum distance loop computation based on the concept of straight line being the shortest distance between any two points in a two dimensional space.

4.2 Two Points and One Circle

In this sub section we consider another version of a minimum distance problem which is quite dealt in detail because it is the crucial element in building the final solution.

Here we have the set S containing three elements $S = \{S_0, S_1, S_2\}$. $S_0 = \{a_{01}\}$ and $S_1 = \{a_{11}\}$ are two points and S_2 is a set of points that forms the circumference of a circle. This is a specific case of a problem that is referred to as ‘milkmaid problem’ [22]. Loosely stated, the milkmaid problem is the problem of finding a point on the river to which the milkmaid, whose position is given, should visit for rinsing her bucket and then go to the cow at another specific location. The cow and the milkmaid are on the same side of the river, which takes the shape of any curvature and that the route taken by the milkmaid is shortest. There are different methods in calculus that is directed towards approaching these problems like Lagrange’s multipliers and Newton’s method. We have our current sub problem Two-Points-and-One-Circle (TPOC) if the curve in the milkmaid problem is a perfect circle.

Thus we are after the minimum collective distance $d_{a_{01} a_{2x}} + d_{a_{2x} a_{11}}$ (d as given in (3)) where a_{2x} is any element of the set S_2 . Here the set S_2 is the set with the elements following a cyclic order as they are the points on the circumference of the circle being represented which are arranged in a cycle.

There is no exact solution to this problem as we had in the earlier case of two points and one line. Geometrically, the non trivial case is only when any one point is outside the tangential space formed by the tangents that enclose the circle by the other point and also when the two points are not in line with the centre of the circle. Otherwise, the trivial solution is the straight line formed that joins the two points. The optimum point on the circle will be any one point of intersection between the circle and the line joining the given two points. In the non trivial case, any one point has to be outside the tangential space formed by the tangents to the circle from the other point. If one of the two points is positioned outside the tangents of the other point enclosing the circle, then the vice versa is also true.

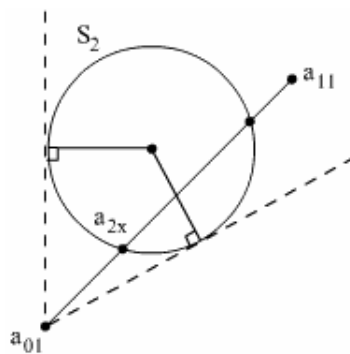


Figure 3. Trivial case

For the non trivial case, finding the best point on the set of points on the circumference can be an exhaustive search. However, rather than having every element of the set S_2 to be considered for a potential candidate, the search space can be very much reduced. It can be reduced to the set of points on an arc segment rather than the points on the entire circumference of the circle traced by the elements (points) of the set S_2 .

Let $a_{2k} \in S_2$ be the point on S_2 that is closest to the point a_{01} and let $a_{2l} \in S_2$, be the closest point to a_{11} . Geometrically, a_{2k} is the point of intersection of the circle traced by the elements of S_2 and the line joining a_{01} and the centre of the circle. Now, the search space is confined to the points that are within the smaller arc formed between the points a_{2k} and a_{2l} on S_2 . If the set S_2 contains a set of points that represents points on the circumference of the circle in a sequence, then the search space is within the closed interval $[a_{2k}, a_{2l}]$ or $[a_{2l}, a_{2k}]$ of the cyclic ordered S_2 . The smaller one is considered as the reduced search space for the optimum solution. The points a_{2k} and a_{2l} are on the circumference of a circle, and they induce two arc segments between them and the short arc alone represents the reduced search space.

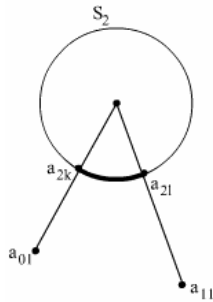


Figure 4. Reduced search space

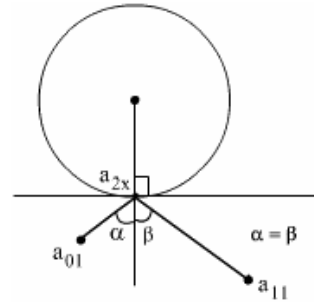


Figure 5. Optimum solution criteria

It can be proved that the small arc between the points a_{2k} and a_{2l} contains the optimum point based on the equal angles towards the normal approach.

Any point on the arc segment could be a candidate for the optimum point and we consider α being the angle between the normal at the candidate point and the line segment joining a_{01} and the candidate point. Similarly, let β be the angle between the normal at the candidate point and the line segment joining a_{11} and the candidate point. As we saw in the previous case, the candidate point a_{2x} will be the optimum point if and only if angles $\alpha = \beta$. This is illustrated in the above figure.

Consider the following figure. At a_{2k} , $\alpha = 0$ and β is some non-zero entity. At the point a_{2l} , $\beta = 0$ and α is non-zero. Thus, tracing every point from the ordered

sequence $\langle a_{2x} \rangle_{x=k}^l$, α increases from zero to a finite value and β reduces from some finite value to zero. Thus, for α to be equal to β , it has to happen somewhere in between a_{2k} and a_{2l} and thus proved that the reduction of search space still retains the optimum point without any loss.

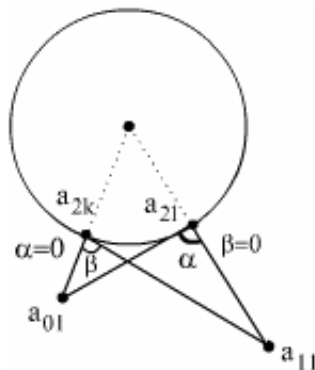


Figure 6. Proof of correctness of reduced search space

It can also be noted that the reduced search space is always less than at least half the size of the original search space and tries to reach half the size as the distance between the points become far. Now, having reduced the search space, there can be numerous methods to find the best point for the optimum solution in the search space. We can use a binary search style method or branch and bound style methods in order to arrive at the optimum point. This Two-points-and-One-Circle procedure for finding the minimum length loop touching all the entities is the key procedure in the instrumentation of the approximation of the final solution for our problem.

4.3 Two Lines and One Point

This sub problem Two-Lines-and-One-Point (TLOP) is covered primarily for the comprehensive coverage of the minimum loop problem of three entities with two being the unknown. This problem also demonstrates the key idea of the heuristic which is the regressive iteration and provides a basic understanding of the execution of the procedure. This problem module involves two lines and a point. Given a point and two lines, the objective is to find a point on each of the two lines such that the perimeter of the triangle formed by these points and the given point is minimal. Again S is the set of three elements $S = \{S_0, S_1, S_2\}$, where $S_0 = \{a_{01}\}$, S_1 and S_2 denote the

points representing the two lines. The objective is to find a_{1x} and a_{2y} on S_1 and S_2 respectively, such that the perimeter of the triangle formed by a_{01} , a_{1x} and a_{2y} is minimum.

This case is quite different from the previous cases because we have to find two points on two sets. It is a more complex case than the previous problems and there is no direct exact solution for the non-trivial case. Also, since S_1 and S_2 represent the discretised points on the lines, these sets contain infinite elements, as discussed earlier in the TPOL problem. However, there is no impact of this issue because the solution points on S_1 and S_2 will not be very far from the projections of the point a_{01} on the lines S_1 and S_2 .

There are two trivial cases for this problem: (a) One trivial case is that the lines S_1 and S_2 are parallel to each other and the point a_{01} is positioned in between them. The solution will contain the points a_{1x} on S_1 which is closest to a_{01} and a_{2y} on S_2 that is closest to a_{01} . Thus, a_{1x} will be the point on S_1 , where the normal drawn from a_{01} to S_1 intersects S_1 and similarly with a_{2y} on S_2 . (b) The other trivial case will be that the point a_{01} is not positioned between the lines S_1 and S_2 . In this case, the solution will be the points of intersection on S_1 and S_2 with the normal drawn from a_{01} to the farthest line. This can be proved again using the triangle inequality.

For the non trivial cases, this problem is harder than the previous ones. We try to achieve the goal of finding the triangle with minimum perimeter with one fixed point and two variable points (on each of the lines) by following an approximation procedure through regressive iterations.

The concept is so very simple but yet amazingly powerful. First, a random point is taken on S_1 and is called a_{1k} . With a_{01} and a_{1k} as the two fixed points and the line S_2 , the best point on S_2 called a_{2l} is found such that the perimeter of the triangle formed by the points a_{01} , a_{1k} and a_{2l} is minimum. This can be effected by using the

heron's procedure for Two-Points-and-One-Line (TPOL). Now, with the newly found point a_{2i} on S_2 and a_{0i} as the two fixed points and S_1 as the line the same procedure is applied again to find the best point in the line S_1 to compute the minimum perimeter. This procedure is repeated again with a_{0i} and the newly found best point on S_1 to find the best point on S_2 . This procedure is iteratively repeated and it can be observed that the total length of the perimeter of the triangle formed in every step reaches to a convergence very quickly. It is also observed in empirical results that the value of the converged loop length is very close and almost the absolute minimum value.

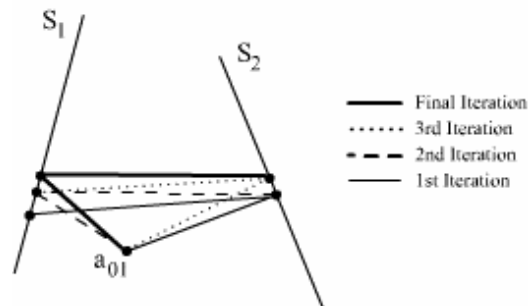


Figure 7. Evaluation of optimum points for Two Lines and One Point

Although we are dealing with circles and points in our original problem, the handling of lines and points will provide a clear understanding on a solid base and will stand for the soundness of the approach which will be later employed in the case of circles and points.

Heuristic for computing the optimal route for Two Lines and One Point

Inputs :

$$S_0 \leftarrow \{a_{0i}\}$$

$$S_1 \leftarrow \{a_{1i}\}_{i=-\infty}^{\infty}$$

$$S_2 \leftarrow \{a_{2i}\}_{i=-\infty}^{\infty}$$

procedure TWO-LINES-ONE-POINT (S_0, S_1, S_2)

```

/* initialisation */
oldPerimeter ← 0
newPerimeter ← 0

```

```

tempPoint1 ← null
tempPoint2 ← null
tempPoint ← null
tempBestPoint ← null

/* pick a random point tempPoint1 from  $S_1$  */
/* perimeter-computes the perimeter of the triangle formed by
given points */
tempPoint2 ← TPOL( $a_{01}$ , tempPoint1,  $S_2$ )
newPerimeter ← perimeter( $a_{01}$ , tempPoint1, tempPoint2)
tempLine ←  $S_1$ 
tempBestPoint ← tempPoint2

while newPerimeter  $\neq$  oldPerimeter do

    oldPerimeter ← newPerimeter
    tempPoint ← tempBestPoint
    tempBestPoint ← TPOL( $a_{01}$ , tempPoint, tempLine)
    newPerimeter ← perimeter( $a_{01}$ , tempPoint, tempBestPoint)
    if tempLine =  $S_1$  then
        tempLine ←  $S_2$ 
    else if tempLine =  $S_2$  then
        tempLine ←  $S_1$ 
end do

/* output - method outputting the best points */
output(tempPoint, tempBestPoint)

```

4.4 Two Circles and One Point

This problem of Two-Circles-and-One-Point (TCOP) is similar to the above problem of TLOP except that the lines are replaced by circles. We now have $S = \{S_0, S_1, S_2\}$, where $S_0 = \{a_{01}\}$ and S_1 and S_2 represent the points on the circumference on the circles. The point is assumed to be outside of both the circles. The objective is to find a point each on the sets S_1 and S_2 , such that the triangle formed by those two points along with a_{01} has the minimum perimeter.

The solution for this problem in the non-trivial case is not easy to compute as in the previous sub problem. The trivial case will be the case where the point a_{01} is exactly on the line connecting the centre points of the circles S_1 and S_2 . The solution would

then be points of intersection on the circumferences of the circles and the line joining their centre points. The other trivial case is depicted in the following figure. If the point is anywhere on the grey area, then the best points on the circles will lie on the line connecting a_{01} and the centre of the farthest circle.

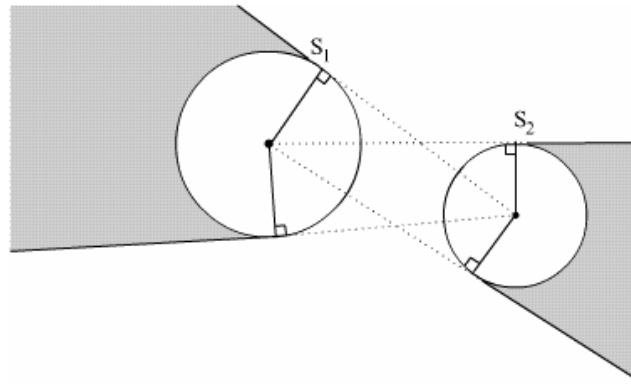


Figure 8. Trivial case for Two Circles and One Point

For the non trivial case, the same procedure employed in the above sub problem can be used in this case as well. A random point a_{1k} from S_1 is taken. With the points a_{01} and a_{1k} and the circle S_2 , the procedure $TPOC(a_{01}, S_2, a_{1k})$ is used to find the best point in S_2 , which we shall call a_{2l} , such that the perimeter of the triangle formed by the points a_{01} , a_{1k} and a_{2l} is minimum. Now, we follow the same methodology of successive and iterative regression that has been used in the earlier problem of TLOP. We compute the new best point on S_1 using the procedure $TPOC(a_{01}, S_1, a_{2l})$, that is again coupled with a_{01} to find the next best value on S_2 . This procedure is repeated until stabilisation is reached with respect to the loop distance. It was seen that the convergence of the loop distance was rather rapid giving the best points absolutely close to the optimal points on S_1 and S_2 . The discrepancies can be attributed to the errors due to discretisation and rounding off, as the experiments were performed on integer based pixel space. The pseudo code of this procedure is congruent to the procedure of Two-Lines-One-Point except in using TPOC instead of TPOL and supplying arguments accordingly.

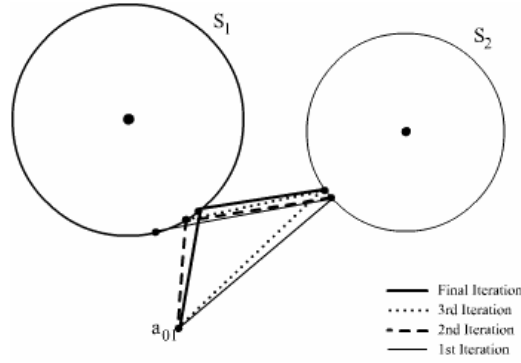


Figure 9. Computation of optimum points for Two Circles and One Point

5. OPTIMAL ROUTE COMPUTATION FOR N CIRCLES

Having built a sound understanding of constructing reasonable solutions for minimum distance loop problems that were discussed, let us proceed to build the solution of the main problem. The problem of computing the optimal route for the mobile sink in the sensor field as described in the initial sections is now trimmed down to be a problem similar to the TCOP problem where there are n circles and one point. We have n circles and a single point in a 2 dimensional space and the objective is to find a point on each of the circle (on its circumference) such that the cumulative length of the loop starting and ending at the fixed point and passing through the points found is minimum.

Given the set S of finite element sets $S_0, S_1, S_2, \dots, S_n$ we have to find a sequence of ordered elements starting and ending with the same element such that there is one element chosen from every set $S_i, \forall i = 0, 1, \dots, n$ and that a operation performed on the sequence gives an optimum result, the operation being the cumulative distance between the points represented by the elements of the sets S_i . As discussed earlier, there will be an exhaustive total of $n!M^n/2$ loops to examine for the shortest loop, where n is the total number of sets with cardinality greater than unity (sets representing circles excluding the one representing the point) and M is the maximum cardinality of the sets $\{S_i\}_{i=1}^n$. This explosive number is due to the combinatorial explosion of the points in the search space. It is very unlikely to even approximate the

optimal solution to this problem unless additional constraints are added to reduce the dimensionality in the complexity of the problem.

One plausible constraint that can be imposed to reduce the hardness is that the order of visiting the sets is known a priori. We proceed to present the method that reaches the approximate solution to the problem, given that the order to visit the sets is known. The decision regarding the order greatly affects the nature of the final solution in terms of its optimality. Thus there are two phases in finding the best possible solution. The first one is the determination of the order of visiting the sets and the second one being computing the optimal points on the sets for the given order. We will focus on the latter part in this section and will reserve the discussions regarding the ordering to the further section where the ordering of sets is analysed in detail and the experimental results are used to compare the relatively better orderings.

Let us assume that the order of visiting the sets is given by the indices on the sets. Thus, S_1 is the first set to visit, followed by S_2 and then the last set will be S_n . We consider the principle of elastics conceptually to derive the best point in every set. Considering the geometric scenario with n circles and one point, we are given the order of visiting the circles. The best point that will be a part of the optimum loop on every circle will be the best point with respect to the best points of the adjacent circles i.e. for the circle $S_i, 1 < i < n$, the best point should be found with respect to the best points of the circles S_{i-1} and S_{i+1} . When $i = 1$, for the circle S_1 , the best point on it is found with respect to the starting point given by the sole element of the set S_0 and the best point on the circle S_2 . Similarly, the best point of the circle S_n is evaluated with respect to the best point on circle S_{n-1} and the starting point $a_{01} \in S_0$. This is like putting a stretched piece of elastic over the circles in order and then allowing it to shrink such that it leaves none of the circles. The shrinking at every circle happens with respect to the adjacent circles. Hence, we employ this approach in order to arrive at a reasonable solution because of the sensibility of the approach.

Now, the task is to find the best points at each of the circles. We extensively use the solution of the sub problem 2 which deals with two points and one circle (TPOC).

Initially, we take one random point from every set S_i which will be $\{a_{ij}\}$, where the domain of the values of i and j are $i = \{0, 1, \dots, n\}$ and $j_i = \{1, 2, \dots, |S_i|\}$.

Thus the initial set of best points will be $\{a_{01}, a_{1j_1}, a_{2j_2}, \dots, a_{nj_n}\}$. Given the procedure for two points and one circle as TPOC(point 1, circle 1, point 2), first we execute, TPOC (a_{01}, S_1, a_{2j_2}) and this will return the best point on circle S_1 with respect to the points a_{01} and a_{2j_2} . This will give a new value for a_{1j_1} that will override the previous value. Now the procedure TPOC(a_{1j_1}, S_2, a_{3j_3}) is run and a new value for a_{2j_2} is found. Finally the execution of the procedure TPOC($a_{n-1j_{n-1}}, S_n, a_{01}$) that gives the new best value of a_{nj_n} will mark the completion of the first iteration. At the end of this iteration we will have a new set of best points $\{a_{01}, a_{1j_1}, a_{2j_2}, \dots, a_{nj_n}\}$ different from the earlier set and that gives a better solution. By ‘better’ solution, we mean that the current loop distance with these points is smaller than the loop distance with the previous set of best points. Now, this iterative procedure is repeated again with the new set of best points that will result in a better set of best points. The iteration is repeated until the loop length stabilises. From the experiments, it can be observed that the stabilisation of the loop length happens fairly quickly within few iterations and thus it can be safely concluded that this procedure does give a very reasonable better approximate solution considering the smaller linear time it takes for execution for a massive problem.

Heuristic for computing the optimal route for n-Circles and One Point

Inputs :

$$S_0 \leftarrow \{a_{01}\}$$

$$S_1 \leftarrow \{a_{1i}\}_{i=1}^{|S_1|}$$

$$S_2 \leftarrow \{a_{2i}\}_{i=1}^{|S_2|}$$

.

.

$$S_n \leftarrow \{a_{ni}\}_{i=1}^{|S_n|}$$

procedure N-CIRCLES-ONE-POINT ($S_0, S_1, S_2, \dots, S_n$)

/* initialisation */

```

oldLoopDistance ← 0
newLoopDistance ← 0

/* Pick random points from  $S_1$  to  $S_n$  & call them  $a_{1j_1}, a_{2j_2}, \dots, a_{nj_n}$  */
/* FindLoopDistance - calculates the length of the loop formed
by  $n+1$  ordered points */
newLoopDistance ← FindLoopDistance( $a_{01}, a_{1j_1}, a_{2j_2}, \dots, a_{nj_n}$ )

while newLoopDistance != oldLoopDistance do

    oldLoopDistance ← newLoopDistance
    for  $i=1$  to  $n$  do

         $a_{ij_i} = \text{TPOC}(a_{i-1j_{i-1}}, S_i, a_{i+1 \bmod n+1 j_{i+1 \bmod n+1}})$ 
    end do
    newLoopDistance = FindLoopDistance( $a_{01}, a_{1j_1}, a_{2j_2}, \dots, a_{nj_n}$ )
end do

/* output - method outputting the best points */
output( $a_{1j_1}, a_{2j_2}, \dots, a_{nj_n}$ )

```

6. ORDERING OF CIRCLES

It has been observed that the above procedure gives an approximate solution fairly quickly for a given order of visiting the circles (sets). Hence, given an order, we have a method that computes a very reasonable answer. However, this is not sufficient for the broader problem that we have defined earlier. The current task is to find the best order for which the final solution becomes better. The order is evaluated with respect to the accuracy that it injects to the final solution as well as the overhead time it takes for determining the order. Ascertaining the ordering of circles is not a trivial task either. There are $n!/2$ orderings possible for n circles because of their cyclic symmetry. This is a huge number and choosing at least a better ordering rather than the best ordering could be certainly rewarding. The notion of ‘better’ is subjective to two principal parameters: precision of the result and the running time for execution. In general, they both are conflicting parameters and typically a trade-off is established between them. In this section we examine few orderings of the circles and analyse their performance with respect to both these contrasting benchmarks experimentally and conceptually.

The most intuitive and direct way to order the circles is by ordering them based on the TSP order of their centre points. This ordering is conceptually similar to approximating the circles to their corresponding centre points. The main issue in this case is the running time for the execution of TSP procedure over the n points. Also, if the circles are of large and varying radii, then the centre points TSP ordering may not provide the best order to visit the circles. This is primarily because of the fact that the inaccuracy of representing a circle with its centre point grows with the circle's size.

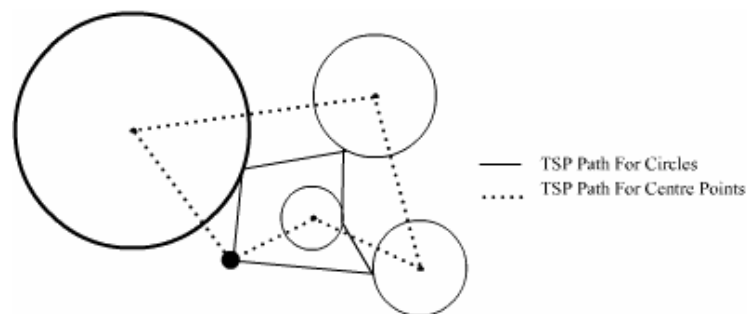


Figure 10. Centre point TSP ordering not being the best ordering

Another default ordering to consider will be random ordering of the circles. This ordering (or rather no ordering) will lift the intensity of the rigorous TSP computation from the procedure but will very likely to provide results that are far from being close to optimum. Nevertheless, it is a method which measures up the comparison of the efficiency of other methods with respect to optimum nature of the solution and running time. Similarly there is another method which was used for experimentation purposes that used the closest points on the circles with respect to the starting point a_{01} , to be fed to the TSP procedure. This is just another variant of uniformly representing all the circles with points on them. This ordering is not of theoretic interest.

It can be intuitively observed that proper ordering of the circles requires the TSP ordering of points on the circles that represent them. Points are chosen from every circle which closely represent the corresponding circles and are then fed into the TSP procedure. Obviously, this means more computation because of the TSP procedure, but the precision of the solution becomes much better.

The choice of points that represent the circles can be done in multiple ways. In the centre point TSP ordering, we generalised all the circles in a common way to their centre points. Having the centre points represent the circles may not yield the best generalisation when the circles have larger radii. Also, instead of having a common representation method for all the considered circles (in terms of identifying points on the circles that represent the very ones) in the plane, we can apply different types of representations for different circles. This carries heavy importance because when we choose a point that represents a circle, we intuitively consider the position of the circle in the entire field, its neighbouring circles, the concentration of the circles in its area, the distribution of circles and also the orientation of the presence of circles in the considered area. These parameters vary for every circle, and thus having a uniform representation of points for all the circles becomes is not appropriate.

Having agreed on the apposite generalisation of the circles (to single points) which may not be uniform, we present another method to choose the points to represent circles that is conceptually clean and better than choosing the centre points of the circles or the closest points on the circles to the starting point. Initially a convex hull is formed for all the circles in the plane. The circles that are present on the vertices of the convex polygon thus formed will be the circles that are on the outmost side. Now, with all the circles, it is required to find a centre point for the whole region, which in general can be the centre of gravity of all the circles in the region. Now, the points that represent the outward circles (present on the vertices of the convex hull formed), will be the closest point on the circle towards the new centre point of the region which is computed. The rest of the circles can have the initial representation which will be their centre points. Thus, the TSP procedure will take the centre points of the circles which do not lie on the convex hull and the closest points (towards the centre of gravity) on the remaining circles.

On careful inspection, it becomes obvious that this method of choosing the points for TSP procedure is better and never worse than the method of having the centre points of all the circles as the input for TSP. When we don't know where exactly the best point will lie on a circle in the final loop, it is best to approximate or represent the circle using the centre point. But, if we know the arcual segment of a circle which will contain the resulting best point for the minimum distance loop, then rather than

representing that circle with its centre point, it is very reasonable to represent the circle with a point along that arcual segment for computing the TSP ordering. Thus this more precise representation of the circles will lead to better TSP ordering of the circles as the procedure's inputs are more relevant. This can be seen in the next figure where the highlighted portions of the outmost circles are the potential segments where the best points for minimum loop computation will be present. Hence this segment is considered for finding a point that represents the circle and the closest point towards the centre of gravity which will definitely lie within the highlighted segment. This method of ordering is currently not implemented and is not considered for experimental analysis. It is discussed here because of its conceptual appeal and the performance analysis of this method along with explorations of more reasonable orders are reserved for further study in this line of research.

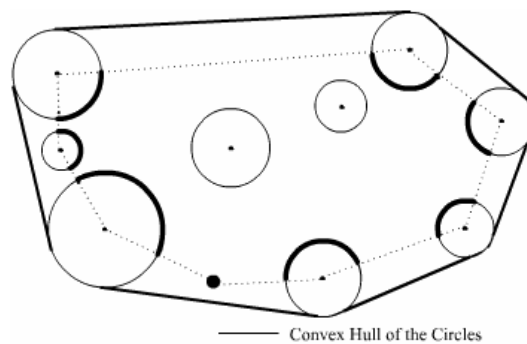


Figure 11. Highlighted arcual segments containing best points of representation

We also propose another ordering of the circles for observing their performance in the experiments. It is more of a heuristic which does not involve heavy computations of the TSP thereby reducing the time taken for processing. However, it pays in the area of optimality as the final result of the minimum distance loop computation is larger than the TSP based ordering of circles. This method orders the circles in a cyclic fashion starting from the top-left quadrant and then sweeping the other quadrants in a circular manner. This method is termed NEWS sort as the circles are ordered based on the direction. The actual outputs of the experimentation are analysed in detail in the following section.

As in any sensor network application, the applications having the discussed scenario of mobile sink and fixed sensor nodes may have certain characteristics that could

impact the design and functioning of the network. It could be a requirement that the order of visiting the nodes is known and fixed. This knowledge can be influenced by the application itself. For instance, the circles are ordered in terms of the ascending order of their radii as the application would require the sink to collect data first from nodes that have smaller transmission range as they could be dying. These external constraints reduce the additional dimension of the problem specification and are not uncommon in the sensor networks scenario.

7. ANALYSIS OF RESULTS

For the purpose of analysing the performance of the heuristic and the various orderings that were described in the earlier sections, a testing platform was developed. This testing platform is part of the Simulated Environment for Networked Sensor Experiments (SENSE) [23]. In addition to the applets provided, individual programs were built to test the performance efficiency of the methods over random samples of problem instances. There were randomly generated circles with radii falling within a preset range in a sample pixel space of 1000 X 1000 grid. The testing environment contains the following: objects such as points and circles; methods for generating random circles; methods that run different algorithms to compute optimum route over the generated circles; and methods that analyse the performance every method and report the results. The following results and graphs are outcomes of the numerical analysis performed over randomly generated data on the testing platform.

7.1 Test Data

The experiments are simulated and are primarily to establish the credibility and the correctness of the approach that we adhered to. Various instances of sample data were created over which the procedures are executed and are observed for their performance. The sample test data is generated random i.e. the circles were randomly generated. Every circle created for a sample has a centre point determined randomly (within the 1000 X 1000 pixel space) and has the value of the radius randomly chosen between 30 and 60 pixels. The starting point (of the sink node) is also randomly selected. In every instance of the problem, there were a fixed number of circles and

there were many samples for every instance. The various instances considered were 3, 4, 10, 25 and 50 number of circles. In order to have an unbiased and generalised view regarding the running time, 100 samples were considered for every instance. For eg. there were 100 samples of 10 randomly generated circles and the starting point, over which the procedures are run and the results are studied. The average of the values of the results is then considered to be the result that closely represents the procedure implementation for the instance containing 10 circles and a starting point. Obviously, the more samples being used, more closer the representation is. For the 3 and 4 circle instances only 5 samples were used because of the intensity of the brute force computation.

7.2 Tested Methods

As discussed earlier, the performance of our procedures are analysed based on two contrasting parameters, running time and accuracy. Having discussed in depth about the computational complexity of our problem, the idea is to find a reasonably precise solution within a reasonable time. The simulated experiments record the minimum distance calculated for every ordering discussed and the time taken by the calculation for every sample. Each sample is fed to four of the mentioned traversal algorithms as input viz. Random ordering, Centre Point TSP, Close Point TSP and NEWS sort. A performance monitor object captures the information regarding the duration of each method over a given sample and the calculated minimum distance covered by the traversal method.

The fundamental objective of the experiments is to derive knowledge about the performance of the various orderings discussed in the previous section. It has been verified that for the TCOP method, the regressive iteration produces answers that are very close to the optimum. We need to check that for n-Circles-and-One-Point which involves the regressive approach along with the elasticity principle, our procedure finds close-to-optimum solution. However, given that the brute force check is far too intensive to compute even for reasonably sized problems, the experiments compute the absolute optimum (minimum distance of the route) only for a given sample of 3 circles and one point which is compared with our method for various orderings. It is

again repeated for samples with 4 circles. The averaged results for 5 samples of these instances are presented in the following table. The distance is calibrated in pixels whereas the duration is measured in milliseconds.

Instance Method	<i>3-Circle</i>		<i>4-Circle</i>	
	<i>Distance</i>	<i>Duration</i>	<i>Distance</i>	<i>Duration</i>
<i>Random</i>	2264	76	2505	65
<i>ClsPnt TSP</i>	1807	62	1934	63
<i>CntPnt TSP</i>	1807	58	1934	65
<i>NEWS sort</i>	1832	54	2169	62
<i>Brute Force</i>	1789	27049	1911	11557996

Table 1. Results of Methods for 3 and 4 circle instances

From the table it can be seen that, for 3 circles the minimum distance computed by our procedures are ‘close’ enough to the absolute minimum distance whose discovery takes a large span of time in comparing with that of the developed procedures. The trend with respect to the value of the distance remains the same for 4 circles as well and it can be safely deduced regarding the continual of that trend for increasing number of circles. Although, with increasing number of circles, the deviation from the absolute minimum may increase slightly, the dramatic increase in the computing time with increasing number of circles will be no match. Hence, these results assure the applicability of our procedure by balancing off the contrasting performance metrics.

7.3 Performance Observations

The approach of iterative regression following the principles of elasticity is verified for any given order. Having established the relevance of the approach of finding the minimum distance for a given order, based on the results which were just discussed, the focus now shifts onto the evaluation of the different orderings. The performance metrics for the same will again be the minimum distance computed and the duration of the execution for every ordering at every instance. The performances of various ordering or traversal methods are pitted against one another and then compared.

7.3.1 Distance

Let us first consider the minimum distance measured by the different ordering methods. As seen for the 3 circles and 4 circles instances, we can use the brute force check to see the deviation of the distances measured by our methods. Quite obviously, the Random ordering method has got the most deviation compared with other methods and the TSP based ordering procedures have the minimal difference from the actual solution. For the 4 circles instance, the deviation of all the methods from the absolute solution is higher than that of the 3 circles instance. However, this increase is sharper in case of Random ordering and the NEWS sort and the increase is very small in the TSP based methods. This is a practical proof of concept that was discussed in the earlier section regarding the orderings based on TSP.

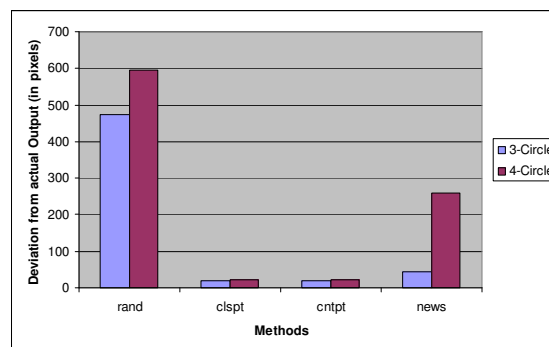


Figure 12. Comparison of outputs' closeness to the solution

Now, in order to analyse the behavioural pattern of the orderings when the number of circles increase, the following graph charts the values of the minimum distance calculated by every method against various instances. It can be noted that Random ordering can easily be the worst case and the TSP based orderings are relatively better. Although the NEWS ordering heuristic provides a better result than the Random ordering, it is still outperformed by the TSP based ones.

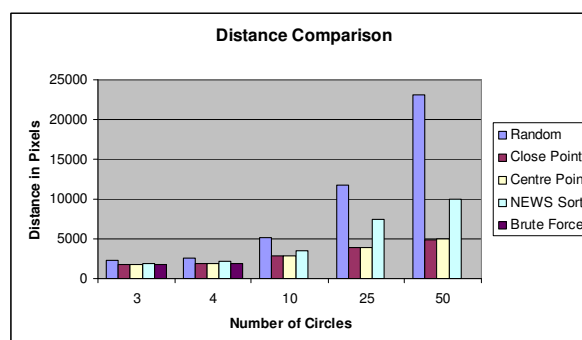


Figure 13. Distance comparison for all the methods

The next graph sketches the data based on the normalised values of the (averaged) distance measures for every instance.

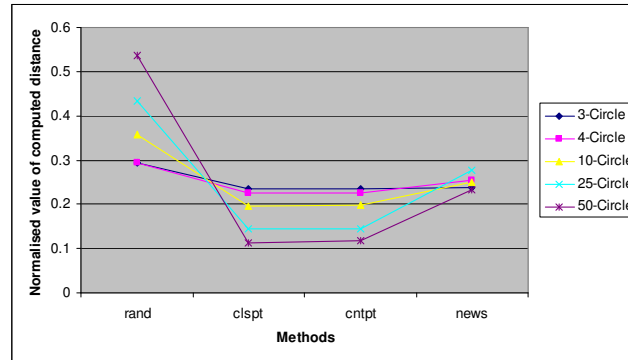


Figure 14. Normalised results of all the instances

It can be seen from the graph that the normalised result of the TSP based orderings becomes smaller when the number of circles increase due to the increase in the normalised values produced by the Random and NEWS orderings. This again ascertains the preferability of the TSP based orderings. Recapitulating the discussion on adaptive representation of a circle (by a suitable point on it), it can now be seen that the method using the convex hull and the centre of gravity to find the representing points will certainly be superior over all these methods experimented, in terms of the accuracy of the calculated distance.

7.3.2 Time

By default, it can be noted that the behaviour of the orderings with respect to the time are opposed to that of the orderings with respect to the accuracy. The methods that measured up very well in terms of accuracy pay in the area of execution time. Let us initially interpret the duration of the program executions visually and then try to balance of the best-worst performance of individual methods on the time-precision scale. The following experimental results are outcomes of tests run on an Intel Pentium 4 2.79GHz Processor with 512MB RAM.

The brute force search for the optimum solution performed for the 3-Circle and 4-Circle problem instances, took a very large time for execution in comparison with the other methods, which is obviously apparent, knowing about the complexity of the

problem. The brute force search method grows from 27 seconds for a 3-Circle sample to 3.21 hours for a 4-Circle sample.

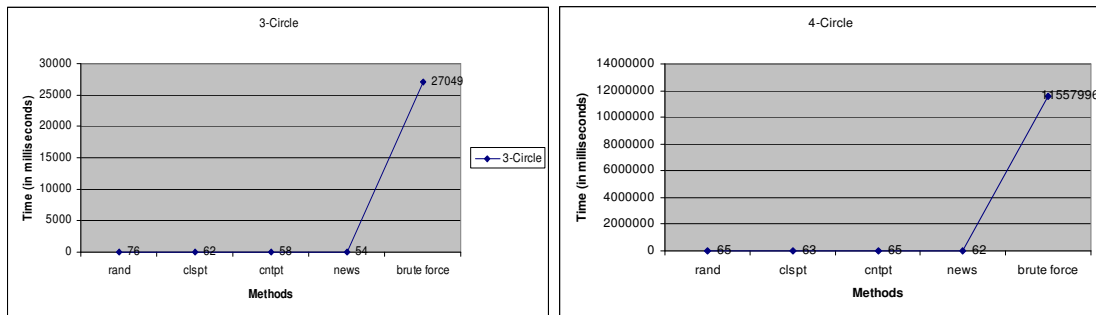


Figure 15. Time taken for all the methods for 3 and 4 circle instances

For a 5-Circle sample, it can be estimated that the execution time will approximately take 39 days. In these cases, where the value which is compared goes beyond reasonable limits, the number of digits of the value is a good measure to compare. The following graph sketches such a comparison. The graph showing the actual time taken by the methods for different number of circles is also presented.

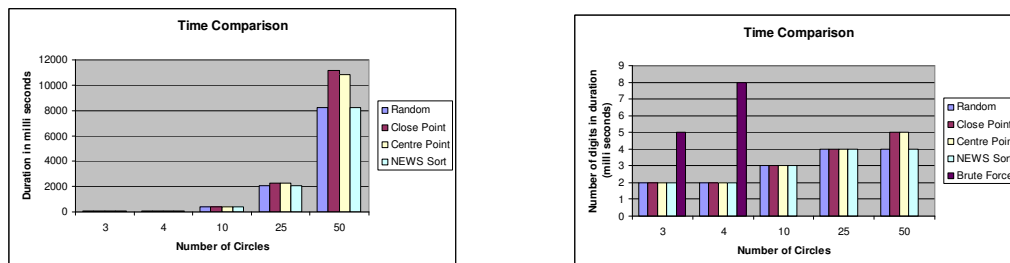


Figure 16. Time comparison based on actual value and number of digits

It can be seen from the above graphs that the methods involving TSP take more time than the Random ordering or the NEWS heuristic because of the absence of any intensive procedures in the latter ones.

Clearly these methods have contrasting behaviour with respect to the time and the accuracy. In order to evaluate the methods by balancing the running time and minimum distance, we consider the best and worst methods with respect to accuracy for a particular problem instance say 50-Circles. The methods are Close Point TSP and the Random ordering respectively. We now plot the rate of increase in the computed

value (rate of decrease of precision) of the Random ordering with respect to the Close point TSP method. Also the rate of decrease in running time for the Random ordering with respect to the other method's running time is plotted.

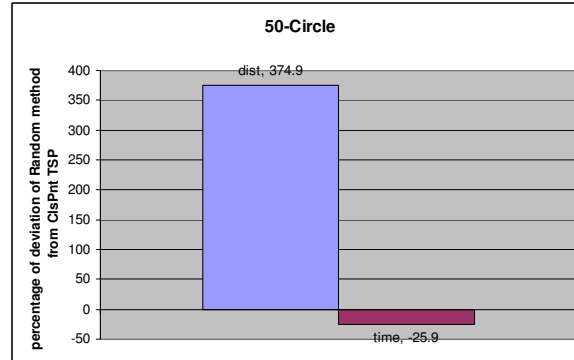


Figure 17. Accuracy and Time deviation between ClsPnt TSP and Random ordering

It can be seen from the graph that for a 50-circle problem sample, the Random ordering method computes the distance that is around 375 percent times the value computed by the Close point TSP method. However, the former method performs about 26 percent faster than the latter method. Yet, since its accuracy is 375 percent worse than the other method, it can be deduced that Close point TSP method is relatively better than the other methods. The inference from the graph is the understanding of the applicability and the superiority of the Close point TSP method (with respect to the other methods discussed), as even though the time taken is relatively longer, the precision is much better than the others.

8. CONCLUSION

Motivated by a practical problem encountered during the design phase of a particular set up of a wireless sensor network, this paper formulates it as a traditional optimisation problem which is subsequently mapped to a classical computational problem. Having discussed the surrounding well performed research, the paper delves into providing a heuristic based on a numerical method approach and the detailed discourse of the subject ensues. The construction of the solution is classified to be of two phases which are individually examined. The suitability of the heuristic is

evaluated with the support of the experimentations that are executed over simulations of test cases.

There are multiple lines of further study in this area which could be interesting as well as contributing to the fine tuning of the current solution. The TPOC procedure is the repetitively used procedure in the computation of the minimal distance route and hence any inherent refinement in that procedure execution will eventually improve the overall method's performance with respect to the time. Also, the investigation of a better ordering than the ones that were discussed in this paper will be a potential area of improvement.

REFERENCES

1. S. Arora, "Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems", *Journal of the ACM (JACM)*, vol. 45(5), pp.753-782, September 1998.
2. M.R. Garey, R.L. Graham and D.S. Johnson, "Some NP-complete geometric problems", In *Proceedings of the 8th Annual ACM Symposium on Theory of Computing*, pp. 10-22, 1976.
3. C.H. Papadimitriou, "Euclidean TSP is NP-complete", *Theoretical Computer Science*, 4, pp. 237-244, 1977.
4. G. B. Dantzig, R. Fulkerson, and S. M. Johnson, "Solution of a large-scale traveling salesman problem", *Operations Research*, 2, pp. 393-410, 1954.
5. N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem", Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.
6. C. H. Papadimitriou and S. Vempala, "On the approximability of the traveling salesman problem (extended abstract)", In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pp. 126-133, 2000.
7. L. Engebretsen and M. Karpinski, "Approximation hardness of TSP with bounded metrics", In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science, Springer-Verlag, vol. 2076, pp. 201-212, 2001.

8. J.E. Goodman and J. O'Rourke, *Handbook of Discrete and Computational Geometry*, CRC Press, 1997.
9. E.M. Arkin and R. Hassin, "Approximation algorithms for the geometric covering salesman problem", *Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 55, pp. 197-218, 1994.
10. J.T. Current and D.A. Schilling, "The Covering Salesman Problem", *Transportation Science*, 23, pp. 208-213, 1989.
11. H. Jonsson, "The traveling salesman problem for lines in the plane", *Information Processing Letters*, vol. 82, 3, pp. 137-142, 2002.
12. C.S. Mata and J.B. Mitchell, "Approximation algorithms for geometric tour and network design problems", In *Proceedings of the 11th Annual ACM Symposium on Computational Geometry*, pp. 360-369, 1995.
13. D.Z. Du, L.Q. Pan and M.T. Shing, "Minimum edge length guillotine rectangular partition", Technical Report MSRI 02418-86, University of California, Berkeley, CA, 1986.
14. J. Gudmundsson and C. Levcopoulos, "A fast approximation algorithm for TSP with neighborhoods", *Nordic Journal of Computing*, vol. 6, 4, pp. 469-488, 1999.
15. M. de Berg, J. Gudmundsson, M.J. Katz, C. Levcopoulos, M.H. Overmars and A.F. van der Stappen, "TSP with neighborhoods of varying size", In *Proceedings of 10th Annual European Symposium on Algorithms*, pp. 187-199, 2002.
16. J. S. B. Mitchell, "Geometric shortest paths and network optimization", In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, Elsevier Science, Amsterdam, pp. 633-701, 1998.
17. J. Hershberger and S. Suri, "A pedestrian approach to ray shooting: Shoot a ray, take a walk", *Journal of Algorithms*, vol. 18, 3, pp. 403-431, 1995.
18. G. Reich and P. Widmayer, "Beyond Steiner's problem: A VLSI oriented generalization", In *Proceedings of 15th International Workshop on Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science, Springer-Verlag, vol. 411, pp. 196-210, 1990.
19. J. Gudmundsson and C. Levcopoulos, "Hardness result for TSP with neighborhoods", Technical Report LU-CS-TR: 2000-216, Department of Computer Science, Lund University, Sweden, 2000.
20. W. Dunham, *Journey Through Genius: The Great Theorems of Mathematics*, Wiley, New York, 1990.

21. M. Kline, *Mathematical Thought from Ancient to Modern Times*, Oxford University Press, New York, 1972.
22. <http://theory.uchicago.edu/~sjensen/teaching/tutorials/Lagrange.html> (accessed on June 29, 2006).
23. <http://www.itee.uq.edu.au/~sense/> (accessed on June 29, 2006).