Range Assignment of Base-Stations Maximizing Coverage Area without Interference

Ankush Acharyya*† Minati De‡§ Subhas C. Nandy*¶

Abstract

This note is a study on the problem of maximizing the sum of area of non-overlapping disks centered at a set of given points in $\mathbb{R}^2$. If the points of P are placed on a straight-line, then the problem is solvable in polynomial time. Eppstein [CCCG, pages 260–265, 2016] proposed an $O(n^{\frac{3}{2}})$ time algorithm, for maximizing the sum of radii of non-overlapping balls or disks when the points are arbitrarily placed on a plane. We show that the solution to this problem gives a 2-approximation solution for the area maximization problem by non-overlapping disks or balls. We also present simulation results. Finally, we propose a PTAS for our problem.

Keywords: Quadratic programming, discrete packing, range assignment in wireless communication, approximation algorithm, PTAS.

1 Introduction

Geometric packing problem is an important area of research in computational geometry, and has wide applications in cartography, sensor network, wireless communication, to name a few. In the disk packing problem, the objective is to place maximum number of congruent disks (of a given radius) in a given region. Toth 1940 [3, 12] first gave a complete proof that hexagonal lattice packing produces the densest of all possible disk packings of both regular and irregular region. Several variations of this problem are possible depending on various applications [2, 12]. In this note, we will consider the following variation of the packing problem:

> **Maximum area discrete packing (MADP):** Given a set of points $P = \{p_1, p_2, \dots, p_n\}$ in $\mathbb{R}^2$, compute the radii of a set of non-overlapping disks $\mathcal{C} = \{C_1, C_2, \dots, C_n\}$, where C_i is centered at $p_i \in P$, such that $\sum_{i=1}^{n} \text{area}(C_i)$ is maximum.

The problem can be formulated as a quadratic programming problem as follows. Let r_i be the radius of the disk C_i. Our objective is:

> Maximize $\sum_{i=1}^{n} r_i^2$
> Subject to $r_i + r_j \leq \text{dist}(p_i, p_j)$, $\forall\, p_i, p_j \in P$, $i \neq j$.

Here, $\text{dist}(p_i, p_j)$ denotes the Euclidean distance of p_i and p_j.

The motivation of the problem stems from the range assignment problem in wireless networks. Here the inputs are the base-stations. Each base-station is assigned with a range, and it covers a circular area centered at that base-station with radius equal to its assigned range. The objective is to maximize the area coverage by these base-stations without any interference. In other words, the area covered by two different base-stations should not overlap. Surprisingly, to the best of our knowledge, there is no literature for the MADP problem. A related problem, namely *maximum perimeter discrete packing* (**MPDP**) problem, is studied recently by Eppstein [4], where the objective is to compute the radii of the disks in $\mathcal{C}$ maximizing $\sum_{i=1}^{n} r_i$ subject to the same set of linear constraints. This is a linear programming problem for which polynomial time algorithm exists [10]. In particular, here each constraint consists of only two variables, and such a linear programming problem can be solved in $O(mn^3 \log m)$ time [9], where n and m are number of variables and number of constraints respectively. In [4], a graph-theoretic formulation of the MPDP problem is suggested. Let $G = (V, E)$ be a complete graph whose vertices V correspond to the points in P; the weight of edge $(i, j) \in E$ ($i \neq j$) is $\text{dist}(p_i, p_j)$, which corresponds to the constraint $r_i + r_j \leq \text{dist}(p_i, p_j)$. They computed the minimum weight cycle cover of G in time $O(mn + n^2 \log n)$ time. Since $m = O(n^2)$ in our case, the time complexity of this algorithm is $O(n^3)$. They further considered the fact that a constraint $r_i + r_j \leq \text{dist}(p_i, p_j)$ is useful if $\delta(p_i) + \delta(p_j) \geq \text{dist}(p_i, p_j)$, where $\delta(p)$ is the distance of the point p and its nearest neighbor in P; otherwise that constraint is redundant. They also showed that the number of useful constraints is $O(n)$, and thus the overall time complexity becomes $O(n^2 \log n)$. They used further graph structure to reduce the time complexity. In $\mathbb{R}^d$, the time complexity of this problem is shown to be $O(n^{2-\frac{1}{d}})$.

It is well-known that if Q is a positive definite matrix, then the quadratic programming problem which minimizes $\tilde{X}'Q\tilde{X}$ subject to a set of linear constraints

*ACM Unit, Indian Statistical Institute.
†ankush_r@isical.ac.in
‡Department of CSA, Indian Institute of Science.
§minati@csa.iisc.ernet.in
¶nandysc@isical.ac.in

$A\tilde{X} \leq \tilde{b}$, $\tilde{X} \geq 0$ is solvable in polynomial time [8]. However, if we present our maximization problem as a minimization problem, the diagonal entries of the matrix Q are all -1 and the off-diagonal entries are all zero. Thus, all the eigen values of the matrix Q are -1. It is already proved that the quadratic programming problem is NP-hard when at least one of the eigen values of the matrix Q is negative [11]. Recently, MADP problem is also shown to be NP-hard [1]. For the minimization version of an NP-hard quadratic programming with n variables and m constraints, an $(1 - \frac{1-\epsilon}{(m(1+\epsilon))^2})$-approximation algorithm is proposed in [6], which works for all $\epsilon \in (0, 1 - \frac{1}{\sqrt{2}})$. The time complexity of this algorithm is $(n^3(m \log \frac{1}{\delta} + \log \log \frac{1}{\epsilon}))$, where δ is the radius of the largest ball inside the feasible region defined by the given set of constraints.

For our MADP problem, a *4-approximation algorithm* is easy to get.

> For each point $p_i \in P$, let $\mathcal{N}(p_i) \in P$ be its nearest neighbor. We assign $r_i = \frac{1}{2}\text{dist}(p_i, \mathcal{N}(p_i))$ for each $i = 1, 2, \ldots, n$. Thus, all the constraints are satisfied. The approximation factor follows from the fact that r_i can take maximum value of $\text{dist}(p_i, \mathcal{N}(p_i))$.

In this note, we first show that if the points in P are placed on a straight line, then the MADP problem can be optimally solved in $O(n^2)$ time. As a feasible solution of the MPDP problem is also a feasible solution of the MADP problem, it is very natural to ask whether an optimal solution of the MPDP problem is a good solution for the MADP problem. We answer this question in the affirmative. We show that the optimum solution for the MPDP problem proposed in [4] is a 2-approximation result for the MADP problem. Finally, we propose a PTAS for the MADP problem.

2 Preliminaries

In a solution of the MADP problem, each disk is centered at some point in P. A solution of the MADP problem is said to be *maximal* if each disk touches some other disk in the solution[1]. From now onwards, by a *solution* of a MADP problem, we will mean it to be a *maximal solution*.

The nearest neighbor of a point $p_i \in P$ is denoted by $\mathcal{N}(p_i) \in P$. Here, a point $p_i \in P$ is said to be a *defining point* of the said solution if it appears on the boundary of some disk in the solution; otherwise it is said to be a *non-defining point*. A *non-defining point* $p_i \in P$ will be covered with a disk C_i centered at point p_i, and its radius r_i is either *equal to* or *less than* $\text{dist}(p_i, q_i)$, where $q_i = \mathcal{N}(p_i)$ is a defining point. In the former case, C_i is said to have *full-radius*, and in the later case, C_i is said to have *part-radius* since the boundary of C_i does not have any point in P. Let us consider a neighbor p_j of the point p_i which has a disk $\mathcal{C}_j$ of radius r_j. We will use the term *residue-distance* to indicate a feasible radius for the disk C_i of length $|\text{dist}(p_i, p_j) - r_j|$, $i \neq j$, if $|\text{dist}(p_i, p_j) - r_j| \leq |\text{dist}(p_i, \mathcal{N}(p_i))|$ (see Figure 1). Thus, the residue-distance of a disk C_i (centered at p_i) is zero if $\mathcal{N}(p_i)$ is a *defining point*. For each full-radius (resp. part-radius) of a disk C_i corresponding to p_i, we define a *full-radius interval (resp. part-radius interval)* of length 2·full-radius (resp. 2·part-radius) whose center lies on p_i.

[1]If a zero-radius disk does not touch any other disk in the solution, it or its neighboring disk can be enlarged to increase the total area in the solution.

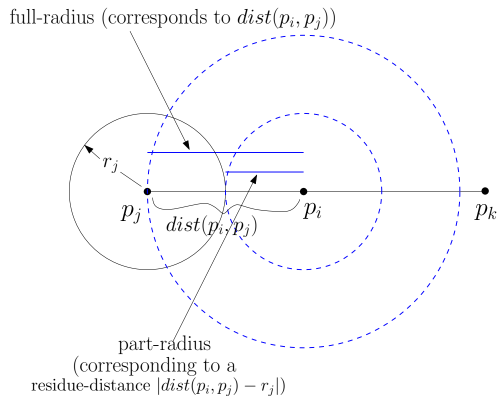

Figure 1: full-radius, part-radius and residue-distance of C_i with respect to p_j

3 MADP problem on a line

In this section we are going to consider a constrained version of the MADP problem, where the point set $P = \{p_1, p_2, \ldots, p_n\}$ lies on a given line L, which is assumed to be the *x-axis*. We also assume $\{p_1, p_2, \ldots, p_n\}$ is sorted in left to right order. Our objective is to place non-overlapping disks centered at each point $p_i \in P$ such that the sum of the area formed by those disks is maximized.

Lemma 1 *In the optimum solution of the MADP problem on a line, at least one of the leftmost or rightmost point in P must be either a* defining point *or its corresponding disk has* full radius.

Proof. Let us denote $d(p_i, p_{i+1}) = d_i$ for all $i = 1, 2, \ldots, n-1$. For the contradiction, let the leftmost point p_1 in P has radius r_1 satisfying $0 < r_1 < \text{dist}(p_1, \mathcal{N}(p_1))$ (see Figure 2). If $r_2 = d_2 < d_1 - r_1$, then we can increase r_1, indicating the non-optimality of the solution. If $r_2 = d_1 - r_1$, then $r_3 = \min(d_3, (d_2 - (d_1 - r_1)))$. Assuming $r_3 = d_2 - (d_1 - r_1)$ and proceeding similarly, we may reach one of the following two situations:

1. $r_k = d_{k-1} - (d_{k-2} - (\ldots(d_1 - r_1)))\ldots)$, and the values of $r_{k+1}, \ldots, r_n$ are independent of r_1.
2. $r_{n-1} = d_{n-2} - (d_{n-3} - (\ldots(d_1 - r_1)))\ldots)$ and $r_n = d_{n-1} - r_{n-1}$.

In Case 1, we show that $S_k = r_1^2 + r_2^2 + \ldots + r_k^2$ can be increased while keeping the values of $r_{k+1}, \ldots, r_n$ unchanged.

$$\begin{aligned} S_k &= \pi \cdot (r_1^2 + (d_1 - r_1)^2 + (d_2 - (d_1 - r_1))^2 + \ldots \\ &\quad + (d_k - (d_{k-1} - (\ldots(d_1 - r_1))))^2) \\ &= \pi \cdot (k \cdot r_1^2 - 2r_1 \cdot c_2 + c_1), \end{aligned}$$

$$\begin{aligned} \text{where } c_1 &= d_1^2 + (d_2 - d_1)^2 + \ldots \\ &\quad + (d_k - (d_{k-1} - (\ldots + (-1)^k \cdot d_1)))^2, \text{ and} \\ c_2 &= (d_1 - (d_2 - d_1) + \ldots \\ &\quad + (-1)^{k-1}(d_k - (d_{k-1} - (\ldots + (-1)^k \cdot d_1))))). \end{aligned}$$

Thus, S_k is a parabolic function whose minimum is attained at $r_1 = \frac{c_2}{k}$, and it attains maximum at the boundary values of the feasible region of r_1, i.e either at $r_1 = 0$ or d_1.

In Case 2, if $r_n > r_{n-1}$, we can increase the sum S_n by setting $r_n = d_{n-1}$, $r_{n-1} = 0$ and keeping $r_1, r_2, \ldots, r_{n-2}$ unchanged. Now, $r_1^2 + r_2^2 + \ldots + r_{n-2}^2$ can further be increased as in Case 1. Similarly, if $r_1 > r_2$ then also S_n can be increased by setting $r_1 = d_1$ and $r_2 = 0$, and then maximizing $r_3^2 + r_4^2 + \ldots + r_n^2$ as in Case 1. If $r_n \le r_{n-1}$ and $r_1 \le r_2$, then also S_n is a parabolic function of r_1, and it is maximized at either $r_1 = 0$ or $r_1 = \min(d_1, \alpha)$ where α = value of r_1 for which $r_{n-1} = d_{n-1}$ [2]. □

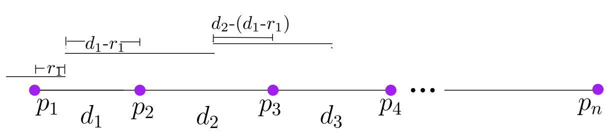

Figure 2: An instance in which $k = 3$

In an optimum solution all the disks have either full-radius or zero radius or has radius equal to the residue distance with respect to the radius of its neighboring points.

Full-radius disks (intervals) are easy to get. For each point p_i, find its nearest neighbor $\mathcal{N}(p_i) = p_{i-1}$ or p_{i+1}, and define an interval of length $2 \cdot \text{dist}(p_i, \mathcal{N}(p_i))$, centered at p_i. We now describe the generation of all possible part-radius intervals for each point $p_i \in P$ considering them in left to right order.

- For both the points p_1 and p_2, there is no part-radius interval.
- If $\mathcal{N}(p_2) = p_1$, then for point p_3, there is a part-radius interval of length $2(d_2 - d_1)$, centered at p_3; otherwise there is no part-radius interval for the point p_3.
- In general, for an arbitrary point p_k if there are m number of part-radius intervals $I_1, I_2, \ldots, I_m$ of lengths $2\delta_1, 2\delta_2, \ldots, 2\delta_m$ respectively, then each of these intervals I_j gives birth to a part-radius interval for the point p_{k+1} with center at p_{k+1} of length $2 \cdot (d_k - \delta_j)$.
  In addition, if $\mathcal{N}(p_k) = p_{k-1}$, then for point p_{k+1}, there is another part-radius interval centered at p_{k+1} and of length $2(d_k - d_{k-1})$.

Finally, we have $\mathcal{I} = \cup_{i=1}^{n} \mathcal{I}_i$. A similar process is performed to generate part-radius intervals $\mathcal{J}$ by considering the points in P in right to left order.

Lemma 2 *For a set P of n points lying on a line L, the maximum number of intervals generated by the above procedure is $\Theta(n^2)$.*

Proof. Let us first consider the forward pass as explained above. Here, for each point p_i (in order) a full-radius interval is generated, and the full-radius interval for point p_i may generate a part-radius interval for each point $p_j, j = i+1, \ldots, n$. Thus, for all the points in P, we may get $O(n^2)$ intervals. To justify the number of intervals is $\Omega(n^2)$, see the demonstration in Figure 3. Here the points $p_i = (x_i, 0)$, $i = 1, 2, \ldots, n$ are placed on the x-axis, where $x_1 = 0, x_2 = 1$ and $x_i = (x_{i-1} - x_{i-2}) + 0.5$, $i = 3, 4, \ldots, n$. Here for each generated interval at p_i, a part-radius interval for the points $p_j, j = i+1, \ldots, n$ will be generated. The same argument follows for the reverse pass also. □

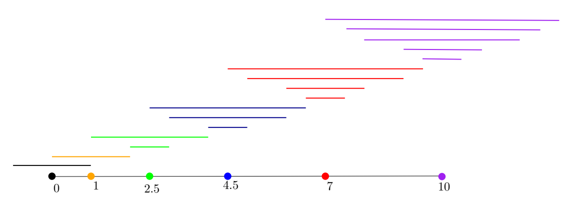

Figure 3: An $\Omega(n^2)$ instance of full and part radius intervals

For each of these intervals we assign weight equal to the square of their half-length. We sort the right end

[2]Here right-end of the feasible region of r_1 is obtained by placing a disk of radius d_{n-1} at p_n, and placing disks at points $p_{n-1}, \ldots, p_2$ touching those of $p_n, \ldots, p_3$, and then placing the disk of radius α at p_1 that touches the disk at p_2. Here surely $\alpha \le d_1$.

points of these intervals. For this sorted set of weighted intervals, we find the maximum weight independent set. This leads us to the following theorem.

Theorem 3 *Given a set P of n points on a line L, one can place non-overlapping disks maximizing sum of their area in $O(n^2)$ time.*

Proof. We can generate the intervals in $O(n^2)$ time as follows. Given a set of intervals $\mathcal{I}_i$ (of full- and part-radius) generated for a point p_i which are sorted by their right end-points, we can generate the set of part-radius intervals $\mathcal{I}_{i+1}$ for the point p_{i+1} in $O(i)$ time. Thus, total time for interval generation is $O(n^2)$ in the worst case. Since intervals for each point p_i are generated in sorted manner, ordering them with respect to their end-points also takes $O(n^2)$ time. Finally, computing the maximum weight independent set of the sorted set of intervals $\cup_{i+1}^{n}\mathcal{I}_i$ using dynamic programming needs $O(n^2)$ time [7].

The correctness of the algorithm follows from the fact that, if there is a interval θ corresponding to point p_i in the optimum solution that does not belong to $\mathcal{I} \cup \mathcal{J}$, then it is not generated by any interval in $\mathcal{I}_{i-1}$ and $\mathcal{J}_{i+1}$. As a result it does not touch any interval of $\mathcal{I}_{i+1}$ and also $\mathcal{J}_{i-1}$. Thus, interval θ can be elongated to increase the total covering area. □

4 Approximation algorithm

In this section, we first show that the optimum solution for the MPDP problem proposed in [4] gives a 2-approximation result for the MADP problem. We also propose a PTAS for the problem.

4.1 2-factor approximation algorithm

Given a set of points P in the plane, let $R = \{r_i, i = 1, 2, \ldots, n\}$ be the set of radii of the points in P obtained by the optimum solution for MPDP problem [4]. It is clear that any feasible solution of the MPDP is a feasible solution of the MADP problem. We show that an optimal solution of the MPDP problem is at most $2 \times OPT$, where OPT is the optimum solution of the corresponding MADP problem.

Lemma 4 *[4] The maximum sum of radii of non-overlapping disks, centered at points $p_i \in P$, equals half of the minimum total edge length of a collection of vertex-disjoint cycles (allowing 2-cycles) spanning the complete geometric graph on the points $p_i \in P$ with each edge having length equal to the distance between the end-points of that edge.*

Lemma 5 *[4] In the minimum total edge length of a collection of vertex-disjoint cycles, each cycle is either of odd length or a 2-cycle (i.e., a single edge).*

The implication of Lemma 4 and 5 is that *in the optimum solution of the MPDP problem, each disk touches its neighboring disk(s) in the cycle in which it appears.*

In [4], an $O(n^{1.5})$ time algorithm is proposed to compute the minimum length cycle cover $\mathcal{C}$ of the complete geometric graph G with a set P of n points on the plane. From the geometric property of the Euclidean distances, they show that if a subgraph G' of G is formed by removing all the edges (p_i, p_j) satisfying $\text{dist}(p_i, \mathcal{N}(p_i)) + \text{dist}(p_j, \mathcal{N}(p_j)) < \text{dist}(p_i, p_j)$, then the minimum weight cycle cover of G' remains same as that in G. We now prove the main result in this section.

Lemma 6 *For a given set of points P arbitrarily placed in the plane, the radii $\{r_i, i = 1, 2, \ldots, n\}$ in the optimum solution of the MPDP problem is a 2-approximation result for the MADP problem for the point set P.*

Proof. As mentioned, MPDP algorithm generates the cycles $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$. We need to show that $\sum_{\alpha=1}^{n} r_\alpha^2 \geq \frac{1}{2}\sum_{\alpha=1}^{n} \rho_\alpha^2$, where ρ_α is the radius in the optimum solution of the MADP problem for the point p_α. We show that $\sum_{p_\alpha \in C_i}^{n} r_\alpha^2 \geq \frac{1}{2}\sum_{p_\alpha \in C_i}^{n} \rho_\alpha^2$ for each cycle $C_i \in \mathcal{C}$. As each disk participates in exactly one of the cycles, aggregating these relations for all the cycles $C_i, i = 1, 2, \ldots, k$, we will have the desired result. Let us consider the following two cases separately.

C_i is a 2-cycle (p_α, p_β): Let $r = \text{dist}(p_\alpha, p_\beta)$. As the disks centered at p_α and p_β in R are touching to each other, let $r_\alpha = \frac{r}{2} - \delta$ and $r_\beta = \frac{r}{2} + \delta$. Thus, $r_\alpha^2 + r_\beta^2 \geq \frac{r^2}{2}$.

Note that in the optimum solution of the MADP problem, the disks for p_α, p_β may not be touching, but $\rho_\alpha + \rho_\beta \leq \text{dist}(p_\alpha, p_\beta)$. So, the upper bound of the sum of squares of the radii in the optimum solution is: $\rho_\alpha^2 + \rho_\beta^2 \leq (\rho_\alpha + \rho_\beta)^2 \leq (\text{dist}(p_\alpha, p_\beta))^2 = r^2$.

Thus, for the two-cycle $\mathcal{C}_i = (p_\alpha, p_\beta)$, we have $r_\alpha^2 + r_\beta^2 \geq \frac{1}{2}(\rho_\alpha^2 + \rho_\beta^2)$.

C_i is an odd cycle: Let the length of the cycle be m. Without loss of generality, assume that the vertices be $p_1, p_2, \ldots, p_m$. For each edge $(p_\alpha, p_{\alpha+1})$ of this cycle (where the indices are numbered modulo m), we have $r_\alpha^2 + r_{\alpha+1}^2 \geq \frac{1}{2}(\rho_\alpha^2 + \rho_{\alpha+1}^2)$ (as explained in the earlier case). Adding these inequalities for $\alpha = 1, 2, \ldots, m$, we have $2\sum_{\alpha=1}^{m} r_\alpha^2 \geq \frac{1}{2}[2\sum_{\alpha=1}^{m} \rho_\alpha^2]$. Ignoring 2 in both sides, we have the result. □

Combining Lemma 6 with the time complexity result in [4], we have the following result.

Theorem 7 *For a given set of points P arbitrarily placed in the plane, one can compute a 2-approximation result of the MADP problem in $O(n^{\frac{3}{2}})$ time.*

4.2 Experimental results

We performed a thorough experimental study on this problem by considering random instances. We considered point sets of different size n, and generated 50 samples[3], where each sample consists of n points. For each sample, we formulated the quadratic programming problem, and run LINDO software to generate the optimum solution MADP$_{opt}$. We also run the MPDP algorithm [4]. Let MADP$_{sol} = \sum_{i=1}^{n} r_i^2$, where $\{r_1, r_2, \ldots, r_n\}$ is the optimum solution of the MPDP problem. For each sample, we computed the ratio $\frac{\text{MADP}_{opt}}{\text{MADP}_{sol}}$, and compute the *average* and *maximum* of these ratios. Finally, we report MADP$_{avg}$ and MADP$_{max}$ for each n. Though, we could only show that the result of the MADP problem using the radii obtained by the MPDP algorithm is a 2-approximation result, it shows much better performance in our experiment on random instances.

Table 1: Experimental result

n	Sum of square of radii obtained by MPDP algorithm	
	average	maximum
10	1.18383	1.5467
20	1.16704	1.38786
30	1.1568	1.39329
40	1.15132	1.18855
50	1.1728	1.23154

4.3 PTAS

In this section, we propose a PTAS for the MADP problem. In [5], Erlebach et al. proposed a $(1+\frac{1}{k})$-approximation algorithm for the maximum weight independent set for the intersection graph of a set of weighted disks of arbitrary size. We will use this algorithm in designing our PTAS.

For each $p_i \in P$, let the maximum possible radius be $\ell_i = \text{dist}(p_i, \mathcal{N}(p_i))$. Thus, the maximum possible area be $\alpha_i = \pi \ell_i^2$. Given an integer k, we compute $h_i = \frac{\alpha_i}{k}$, and define $k+1$ circles $\mathcal{C}_i = \{C_0^i, C_1^i, \ldots, C_k^i\}$ centered at p_i with area $\{0, h_i, 2h_i, \ldots, kh_i\}$ (see Figure 4). Each disk is assigned weight equal to its area. Now we consider all the disks $\cup_{i=1}^{n} \mathcal{C}_i$, and use the algorithm of [5] to compute the maximum weight independent set (MWIS) $\mathcal{A}$. Note that the number of disks centered at any point p_i present in both the optimum solution and in our algorithm for the MWIS problem of $\cup_{i=1}^{n} \mathcal{C}_i$ is exactly one.

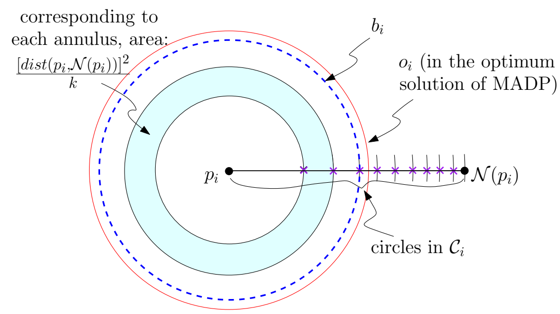

Figure 4: Demonstration of PTAS

Let o_i and a_i be the disks centered at p_i in the optimum solution and in our solution ($\mathcal{A}$) respectively, and O_i, A_i be their respective area. Let $\Theta = \sum_{i=1}^{n} A_i$ be the solution obtained by our algorithm, and $OPT = \sum_{i=1}^{n} O_i$ be the value of the optimum solution. We need to analyze the bound on $\frac{OPT}{\Theta}$.

Let $\widetilde{OPT}$ be the optimum solution of the MWIS problem among the set of disks $\cup_{i=1}^{n} \mathcal{C}_i$. Thus, $\frac{OPT}{\Theta} = \frac{OPT}{\widetilde{OPT}} \times \frac{\widetilde{OPT}}{\Theta}$. Following [5], $\frac{\widetilde{OPT}}{\Theta} \le 1 + \frac{1}{k}$. It remains to analyze $\frac{OPT}{\widetilde{OPT}}$.

Now, let us consider the disks in OPT. For each point p_i, let b_i be the largest disk in $\mathcal{C}_i$ among those which are smaller than equal to o_i (see the blue and red disks in Figure 4). Thus, $\{b_1, b_2 \ldots, b_n\}$ is a feasible solution. Let $LB(OPT) = \sum_{i=1}^{n} B_i$, where B_i = area of the disk b_i. $LB(OPT)$ is the lower bound of OPT.

$\frac{OPT}{\widetilde{OPT}} = \frac{OPT}{LB(OPT)} \times \frac{LB(OPT)}{\widetilde{OPT}}$. Since $\widetilde{OPT}$ is the optimum solution among the disks $\cup_{i=1}^{n} \mathcal{C}_i$, and $LB(OPT)$ is a feasible solution of the MWIS problem among the disks $\cup_{i=1}^{n} \mathcal{C}_i$, we have $\widetilde{OPT} \ge LB(OPT)$.

Now, consider $OPT - \widetilde{OPT} \le OPT - LB(OPT) = \sum_{i=1}^{n} (O_i - B_i) \le \frac{1}{k} \sum_{i=1}^{n} \ell_i^2$, since $O_i - B_i \le \frac{1}{k} \ell_i^2$ by our construction (see Figure 4). We also have $OPT \ge \frac{1}{4} \sum_{i=1}^{n} \ell_i^2$ from the method of getting the 4-approximation result, mentioned in Section 1.

Thus, $\frac{OPT - \widetilde{OPT}}{OPT} \le \frac{4}{k}$, implying $\frac{\widetilde{OPT}}{OPT} \ge 1 - \frac{4}{k}$.

In other words, $\frac{OPT}{\widetilde{OPT}} \le 1 + \frac{1}{k'}$, where $k' = \frac{k-4}{4}$. Thus, $\frac{OPT}{\Theta} \le (1 + \frac{1}{k})(1 + \frac{1}{k'}) \le (1 + \frac{1}{k''})$, where $k'' = \frac{k-4}{5}$. Thus, we have the following result.

[3]Since generating the optimum result is time consuming, the table entries for $n = 40$ and 50, the average and maximum is computed only for 5 samples.

Theorem 8 *Given a set of points P in $\mathbb{R}^2$ and a positive integer k, we can get a $(1 + \frac{1}{k})$-approximation algorithm with time complexity $(nk)^{O(k^2)}$.*

5 Summary

Following Eppstein's work [4] on placing non-overlapping disks for a set given points on the plane to maximize perimeter, we tried to study the area maximization problem under the same setup. We observe that the solution of the perimeter maximization problem gives a 2-approximation result of the area maximization problem. Though the perimeter maximization problem is polynomially solvable, the area maximization problem is NP-hard [1]. However, the said problem has a PTAS. Needs to mention that, if the points are placed on a straight line, then the area maximization problem is solvable in polynomial time.

References

[1] Ankush Acharyya, Minati De, Subhas C Nandy, and Bodhayan Roy. Range Assignment of Base-Stations Maximizing Coverage Area without Interference. *arXiv preprint arXiv:1705.09346*, 2017.

[2] Cédric Bentz, Denis Cornaz, and Bernard Ries. Packing and covering with linear programming: A survey. *European Journal of Operational Research*, 227(3):409–422, 2013.

[3] Hai-Chau Chang and Lih-Chung Wang. A Simple Proof of Thue's Theorem on Circle Packing.

[4] David Eppstein. Maximizing the Sum of Radii of Disjoint Balls or Disks. In *Proceedings of the 28th Canadian Conference on Computational Geometry, CCCG 2016, August 3-5, 2016, Simon Fraser University, Vancouver, British Columbia, Canada*, pages 260–265, 2016.

[5] Thomas Erlebach, Klaus Jansen, and Eike Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM Journal on Computing*, 34(6):1302–1323, 2005.

[6] Minyue Fu, Zhi-Quan Luo, and Yinyu Ye. Approximation Algorithms for Quadratic Programming. *J. Comb. Optim.*, 2(1):29–50, 1998.

[7] Jon Kleinberg and Eva Tardos. *Algorithm design*. Pearson Education India, 2006.

[8] S. P. Tarasov; L. G. Khachiyan M. K. Kozlov. Polynomial solvability of convex quadratic programming. *Doklady Akademii Nauk SSSR*, 248:1049–1051, 1979.

[9] Nimrod Megiddo. Towards a Genuinely Polynomial Algorithm for Linear Programming. *SIAM J. Comput.*, 12(2):347–353, 1983.

[10] Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1982.

[11] Panos M. Pardalos and Stephen A. Vavasis. Quadratic programming with one negative eigenvalue is NP-hard. *J. Global Optimization*, 1(1):15–22, 1991.

[12] Gábor Fejes Tóth. Packing and Covering. In *Handbook of Discrete and Computational Geometry, Second Edition.*, pages 25–52. 2004.