Efficient Beacon Deployment for Large-scale Positioning

Wei He[1,2], Pin-Han Ho[2,3], Dong Wang[1], Lizhong Xiao[1]
[1] School of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai, China
[2]School of Electronics and Information, Nantong University, Jiangsu, China
[3]Department of Electrical and Computer Engineering, University of Waterloo, Canada

Instant and precise localization of indoor mobile users is fundamental for supporting various sophisticated location-aware services. Using Bluetooth low-power beacons for mobile user positioning has been reported as an effective approach, where the beacon deployment positioning (BDP) problem has been defined. The paper introduces a novel approach for solving large-scale BDP problems, aiming to significantly reduce beacon consumption from existing solutions with much less computation complexity. Extensive simulations are conducted to verify the proposed algorithm, whose beacon consumption is about 1.14 to 1.67 times and 0.2 to 0.48 times compared to those of the Mixed Integer Linear Program (ILP) and a naive iBeacon solution respectively. We have also observed that the running time scales well with the growth of the number of Test Positions and attenuation factors.

***Index Terms*—BLE beacon, positioning algorithm, heuristic, ILP, entropy**

I. Introduction

INSTANT and precise localization of mobile users is fundamental for enabling various sophisticated location-aware services, such as guided parking [1], e-fence for sharing bikes [2], Ads and media content distribution [3], guided tour [4] and even location-aware sentimental analysis [5], that are generally in the scopes of the Smart Home [6] and Smart City [7].

Bluetooth Low Energy (BLE) beacons promoted by Google and Apple etc., are known to allow for user-friendly and power-saving deployment to achieve the above goals. In China, some BLE beacons are associated with WeChat applications [8]. In [9], the BDP problem is systematically formulated as a Mixed Integer Linear Program to differentiate any two Test Positions (TPs) in different Shared Information Test Position Groups (SIPGs). The user location is obtained by analyzing the collected "beacons" at each user device. However, solving the ILP and obtaining the optimal beacon deployment is feasible only in small systems.

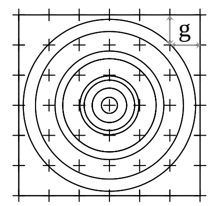
To meet the challenges imposed by large scale positioning problems, the paper introduces a novel heuristic algorithm to achieve a graceful compromise between the optimality and computation complexity. Besides, the proposed heuristic algorithm bears a rather concise solution representation to speed up the localization decoding.

The main contribution of the proposed BDP solution approach is in two-fold. Firstly, the proposed heuristic is equipped with a novel method based on sparse matrix and encoding to reduce the memory usage of BDP solutions. Secondly, a post-processing procedure is introduced to further shrink the beacon consumption on a given BDP solution that is generally applicable to any beacon deployment algorithm.

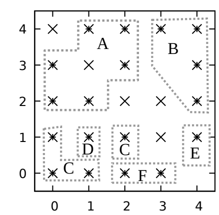
The rest of the paper is organized as follows. Section II discusses the related work. Section III reviews the BDP and presents the problem formulation, including feasibility condition of the BDP problem, definition of the entropy and information criteria useful to quantify the problem solutions, together with the upper-bounds on the positioning accuracy of solutions. Section IV elaborates the proposed heuristic algorithm for beacon deployment and the algorithm for compressing beacon consumption from existing solutions. Section V evaluates the performance of the heuristic algorithm in terms of beacon consumption and running time. Section VI concludes the paper.

lev	pwr (dBm)	$rssi$ (dBm)
0	-30	-91
1	-20	-81
2	-16	-76
3	-12	-74
4	-8	-68
5	-4	-66
6	0	-62
7	4	-60

(a) power level settings



(b) circular radiation patterns



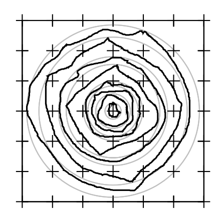
(c) BDP problem

SIPG	pAC	TPs
A	001	(0,2)(0,3)(1,2) (1,4)(2,3)(2,4)
B	010	(3,4)(4,2)(4,3)(4,4)
C	011	(0,0)(1,0)(0,1)(2,1)
D	100	(1,1)
E	101	(4,1)
F	110	(2,0)(3,0)

(d) BDP's pseudo ACT

SIPG	pAC	TPs
(0,2)	00001	(0,2)
(0,3)	00010	(0,3)
(1,2)	00011	(1,2)
(1,4)	00100	(1,4)
(2,3)	00101	(2,3)
(2,4)	00110	(2,4)
(3,4)	00111	(3,4)
(4,2)	01000	(4,2)
(4,3)	01001	(4,3)
(4,4)	01010	(4,4)
(0,0)	01011	(0,0)
(1,0)	01100	(1,0)
(0,1)	01101	(0,1)
(2,1)	01110	(2,1)
(1,1)	01111	(1,1)
(4,1)	10000	(4,1)
(2,0)	10001	(2,0)
(3,0)	10010	(3,0)

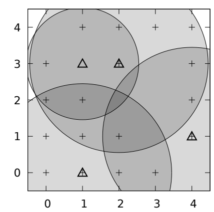
(e) BDUP's pseudo ACT



(f) arbitrary radiation patterns

Fig. 1: AOI: 30*30m, grid gap= $6m$, $\alpha = 3$ [10]

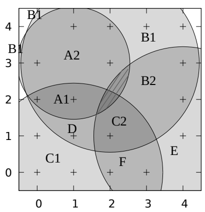
Manuscript received Oct. 3, 2020, revised Mar. 5, 2021;
Corresponding author: Pin-Han Ho (email: p4ho@uwaterloo.ca), Wei He (email: heweist@gmail.com)
Supported by the Natural Science Foundation of Shanghai (Grant No. 20ZR1455600), and the Introduction of Talents Program of Shanghai Institute of Technology (Grant No. YJ2019-1)



(a) The heuristic BDP solution

SIPG	CPs with bcns (1,0)	(4,1)	(1,3)	(2,3)	TPs in AG	AA
A	1	0	1	1	(0,2)(1,2)	A_1
	0	0	1	1	(0,3)(1,4)(2,3)(2,4)	A_2
B	0	0	0	1	(3,4)(4,4)	B_1
	0	1	0	1	(4,2)(4,3)	B_2
C	1	0	0	0	(0,0)(1,0)(0,1)	C_1
	1	1	0	1	(2,1)	C_2
D	1	0	0	1	(1,1)	D
E	0	1	0	0	(4,1)	E
F	1	1	0	0	(2,0)(3,0)	F
lev	6	6	4	6		

(b) The Area Code Table (ACT)



(c) AAs in solution

SIPG	CPs with bcns (1,0)	(4,1)	(1,3)	(2,3)	TPs in AG
A	x	0	1	1	(0,2)(0,3)(1,2)(1,4)(2,3)(2,4)
B	0	x	0	1	(3,4)(4,2)(4,3)(4,4)
C	1	0	0	0	(0,0)(1,0)(0,1)
	1	1	0	1	(2,1)
D	1	0	0	1	(1,1)
E	0	1	0	0	(4,1)
F	1	1	0	0	(2,0)(3,0)
lev	6	6	4	6	

(d) simplified ACT

Fig. 2: Solutions to the problem defined by Fig. 1

II. Related Work

Various approaches have been proposed to improve the accuracy of BLE positioning and extend its applicability to numerous scenarios. Anchor-based positioning algorithms exploit the anchor information such as Received Signal Strength Indicator (RSSI), Angle of Arrival (AoA), Time of Arrival (ToA), and/or environmental data including temperature, air pressure or humidity, to pinpoint users' location. [11] adopts Baysian Filtering to refine collected positioning data. [1] attempts to improve positioning quality in smart parking through particle filter. [12] devises a semi-automatic system capable of BLE-related parameters tuning to achieve high accuracy. [13] explores using software-defined radio to realize a reconfigure positioning system. [14] studies to deliver dynamic content Ads in multiple protocols with BLE beacons. [15] presents a system in which smartphones help collect data from nearby BLE smart-objects and upload to a backend repository.

The cell-based method [16] localizes a user merely based on the SSIDs of beacons received by the user without complex equation solving. To our knowledge, [9] is the first work takes the cell-based method and provides an ILP formulation for the BDP problem along with its theoretical bounds on beacon consumption, where the beacons with arbitrary coverage patterns and multiple power-levels are considered. Nevertheless, the approach of [16] has to deploy a large number of TPs to achieve high precision,which increases the problem scale significantly.

[17] attempts to solve the Bounded Beacon Depolyment Positioning (BBDP) problem via an ILP where the positioning signals are strictly controlled within the given AOI due to security concerns or applications' explicit requirements. Large scale BBDP problems can be divided into multiple BBDP subproblems to be solved independently and concurrently.

Some other works research on hierarchical, distributed or collaborative positioning framework. [18] introduces a layered fingerprinting positioning system utilizing both Wifi and BLE beacons. [19] discusses distributed localization when many users have no direct access to anchors (e.g. BLE beacons). A cooperative linear distributed iterative solution based on local measurements, communication, and computation is proposed. [20] overviews collaborative localization in 5G and IoT applications, and examines its theoretical limits, algorithms, and challenges.

III. Problem Formulation

A. Problem Review

In an *Area of Interest* (AOI), *Test Positions* (TPs), denoted by $\mathbb{T}$ and *Candidate Positions* (CPs), denoted by $\mathbb{C}$, are predefined locations in an AOI where positioning are required and beacons can be installed respectively. For simplicity, TPs and CPs are distributed on equally-spaced grid locations; the distance between any two adjacent rows or columns, denoted by g is called the *grid gap* (See Fig. 1b). Each group of TPs desired to retrieve identical *location-aware* information is defined as a *Shared Information Test Position Group* (SIPG), denoted by $\mathbb{G}$. Take Fig. 1c for example, all TPs and CPs are marked by '$+$' and '$\times$' respectively while SIPGs are shown by dashed polygons enclosing its TPs.

Each beacon has a set of configurable power-levels, denoted by $\mathbb{V}$. For a given AOI, a *Beacon Deployment Positioning* (BDP) problem aims to differentiate any two TPs from different SIPGs by installing least number of beacons on CPs with proper settings. Note that two TPs are *differentiated* if covered by different sets of beacons. Specifically, a *Beacon Deployment Unambiguous Positioning* (BDUP) problem is a BDP problem when each TP composes an SIPG.

Fig. 1a-1c present a BDP problem in an AOI of $30m$ by $30m$ with a grid gap of $6m$. Estimote beacons are adopted with power-level settings given by Fig. 1a [9] . Applying the Log Loss Radio Propagation model [21] with attenuation factor set to 3, Fig.1b depicts the corresponding circular radiation patterns for each power-level. Due to multi-path effects and other factors, real radiation patterns can be of arbitrary shape while bounded by their outer-circles (drawn in gray in Fig. 1f).

Thus, the *radiation range* of an arbitrary radiation pattern is defined as the radius of its outer circle. Correspondingly, the *minimum* and *maximum* radiation range for a beacon, denoted by R_{min} and R_{max}, are the radiation range for the lowest and highest power-consuming power-levels respectively. As in [9] and [17], the number of TPs a beacon can cover at the highest power-level is known as its *density*, denoted by ϕ.

For instance, the power-levels shown in Fig. 1b have $\phi = 25$, $R_{min} = 0.26g$ and $R_{max} = 2.85g$ meters. Similarly, the ones in Fig. 1f have $\phi = 21$ while R_{min} and R_{max} stay the same as the outer-circles are identical for radiation patterns in Fig. 1b and Fig. 1f.

Fig. 2a shows an optimal solution obtained via the proposed heuristic algorithm to the problem defined in Fig. 1a-1c where each beacon is marked by a $\triangle$ and its radiation pattern drawn by a shaded circle. Equivalently, the solution in Fig. 2a can be shown by an *Area Code Table* (ACT). As Fig. 2b shows, the *last row* in an ACT records the power levels configured for the installed beacons. All other rows have 4 columns: the first stores an SIPG's name, the second records an *Area Code* (AC) taken by that SIPG, and the third shows the TPs sharing that AC, forming an *Ambiguity Group* (AG). Note that an SIPG can be split into multiple AGs with distinct ACs.

Geometrically, the area(s) covered by the same set of beacons in a solution form an *Ambiguous Area* (AA), whose name is optionally stored in the fourth column. As demonstrated by Fig. 2c, some AA as B_1 includes multiple non-contagious sub-areas and others with no TPs (like the hatched ones) are ignored in the solution.

An *Area Code* is a binary representation of the signal coverage status by the deployed beacons, each assigned a bit. The bit for a beacon b_i is set to 1 in an AC if the TP(s) in the third column are covered by b_i and 0 otherwise; all bits for b_i compose a bit column, known as the *Beacon Code* for b_i.

The *do not care symbol 'x'* suggested in [22] can be applied to simplify the ACT representation. For each SIPG, two ACs differ by a bit b_i can be combined into one AC whose b_i bit is marked as 'x'. In general, *if all combinations of n bits appeared in an SIPG's ACs, these n bits can be marked by n x's*. For example the ACT in Fig. 2b can be simplified to the one in Fig.2d.

Note that *ACTs can also stipulate the differentiation requirements of BDP problems*. For instance, Fig. 1d creates a peudo ACT as an ideal reference solution corresponding to the problem defined in Fig. 1a-1c where each SIPG is assigned a unique minimal length "peudo" area code; Fig. 1e shows the pseudo ACT to its corresponding BDUP problem.

B. Problem Feasibility

Definition 1. *A pair of TPs are* ***differentiated*** *if covered by different set of beacons; a* ***TP is identified*** *if it is differentiated with all TPs in other SIPGs; an* ***SIPG or AG is identified*** *if all TPs in it are identified.*

Definition 2. *A BDP problem is feasible,* ***iff*** *each TP can be assigned a non-zero AC, and any two TPs from different SIPGs have distinct ACs.*

According to the definition of an AC, non-zero ACs guarantee that all TPs are covered. Since any pair of TPs in different SIPGs have distinct ACs, they can be differentiated by at least a beacon covering only one of them. Therefore, there is always a solution to the given BDP problem.

Theorem 1. *Suppose $\mathbb{T} \subseteq \mathbb{C}$ and a beacon with minimal power-level can cover only one TP. A* ***partial solution is guaranteed feasible*** *if for any undifferentiated TPs t_1 and t_2 from different SIPGs,* ***at most*** *one beacon is installed at either t_1 or t_2, which covers both TPs.*

Proof. When $\mathbb{T} \subseteq \mathbb{C}$, a TP is also a CP. Consider any undifferentiated TPs t_1 and t_2 from different SIPGs. 1) If there are beacons installed at t_1 and t_2 not covering both TPs, t_1 and t_2 are differentiated, contradicting t_1 and t_2 are undifferentiated TPs; 2) If beacons installed at t_1 and t_2 cover each other, the solution is *infeasible* when all other CPs which can differentiate t_1 and t_2 are already used; 3) If at most one beacon is installed at t_1 or t_2 covering both TPs, t_1 and t_2 can safely be differentiated by installing a beacon with minimal power-level at the unused CP from $\{t_1, t_2\}$. □

C. Positioning Accuracy

In this paper, the **positioning accuracy** of a solution is defined as the longest distance between the actual position and the identified position given by the solution.

Theorem 2. *For arbitrary radiation patterns, the positioning accuracy achieved by a feasible solution is upper-bounded by $2R^*_{max}$, where R^*_{max} is the largest radiation range of all deployed beacons in the solution.*

Proof. Note that positions are differentiated through the beacons covering them. Any two positions farther than $2R^*_{max}$ would not be covered by the same beacon and are differentiated. Thus, an accuracy of $2R^*_{max}$ is guaranteed. □

Theorem 3. *For arbitrary radiation patterns, the positioning accuracy achieved by a feasible solution is upper-bounded by the largest diameter of the outer-circles of all ambiguous areas in the solution.*

Proof. Note that the outer-circle of each AA encloses all its sub-area(s) and its diameter imposes an upper-bound on the positioning accuracy. Take the AA F in Fig. 2c for example, a user in any position inside F will be reported in (2,0) or (3,0). Since F is certainly inside a circle with a diameter of $3g$ meters, the positioning accuracy is at least upper-bounded by $3g$. Considering all such outer-circles, the one with largest diameter determines the worst-case accuracy. □

D. Evaluating Solutions and Problems

Inspired by our work in [23], for an ACT $\mathcal{A}$, let $|\mathcal{A}|$ denote the number of ACs, also AGs, in $\mathcal{A}$; then **entropy** of $\mathcal{A}$:

$$entropy(\mathcal{A}) = -\sum_{i=0}^{|\mathcal{A}|-1} p_i \log p_i$$

where p_i represents the frequency of the i_{th} AC appeared in $\mathcal{A}$. Correspondingly, **information** of $\mathcal{A}$:

$$info(\mathcal{A}) = |\mathbb{T}| \cdot entropy(\mathcal{A})$$

. In an ideal solution, each SIPG is assigned one AC as Fig. 1d shows, $entropy(\mathcal{A})_{ideal} = -\sum_{i=0}^{|\text{SIPG}|-1} s_i \log s_i$ where $s_i = \frac{|SIPG_i|}{|\mathbb{T}|}$ and $info(\mathcal{A})_{ideal} = |\mathbb{T}| \cdot entropy(\mathcal{A})_{ideal}$.

For example, suppose the ACTs in Fig. 1d and Fig. 2b are called $\mathcal{A}_0$ and $\mathcal{A}_1$ respectively. Then

$$\begin{aligned} entropy(\mathcal{A}_0) = &-\frac{6}{18}\log_2\frac{6}{18} - 2(\frac{4}{18}\log_2\frac{4}{18} + \frac{1}{18}\log_2\frac{1}{18}) \\ &-\frac{2}{18}\log_2\frac{2}{18} = 2.31\text{bits} \end{aligned} \tag{1}$$

and $info(\mathcal{A}_0) = 18 \cdot entropy(\mathcal{A}_0) = 41.58$bits while

$$\begin{aligned} entropy(\mathcal{A}_1) = &-4(\frac{2}{18}\log_2\frac{2}{18}) - \frac{4}{18}\log_2\frac{4}{18} \\ &-\frac{3}{18}\log_2\frac{3}{18} - 3(\frac{1}{18}\log_2\frac{1}{18}) = 3.02\text{bits} \end{aligned} \tag{2}$$

and $info(\mathcal{A}_1) = 18 \cdot entropy(\mathcal{A}_1) = 54.36$bits.

Now if the problem defined in Fig. 1a-1c is changed to a BDUP problem when each TP under consideration makes an

SIPG. Then, for the ACT shown in Fig. 1e, denoted by $\mathcal{A}'$, $entropy(\mathcal{A}') = -18 \cdot \frac{1}{18} \log_2 \frac{1}{18} = 4.17$bits and $info(\mathcal{A}') = 18 \cdot entropy(\mathcal{A}') = 75.06$bits.

Note that *a problem with more information is harder to solve* since it requires to distinguish more TP pairs. For instance, the information for a BDUP problem shown by Fig. 1e is much larger than the corresponding BDP problem represented by Fig. 1a to Fig. 1c.

On the other hand, *Entropy is handy in evaluating the progress of a partial solution*. For BDUP problems, $entropy(\mathcal{A})_{ideal} = \log_2 |\mathbb{T}|$ is exactly the maximum entropy achieved when BDUP requirements are met; for a BDP problem, it is certainly *unsolved* when $entropy(\mathcal{A}) < entropy(\mathcal{A})_{ideal}$, but $entropy(\mathcal{A}) \geq entropy(\mathcal{A})_{ideal}$ does not imply that the BDP problem is solved since entropy can also increase due to unnecessarily assigning multiple AGs to an SIPG.

To evaluate the differentiation contribution made by a newly added beacon b_i, **entropy gain** can be computed. Let the ACTs before and after adding b_i to a partial solution be denoted by $\mathcal{A}_{old}$ and $\mathcal{A}_{old|b_i}$ respectively. Then, the entropy gained via adding b_i equals to:

$$entropy(\mathcal{A}_{old|b_i}) - entropy(\mathcal{A}_{old})$$

. Clearly, b_i can help reduce the ambiguity only when non-zero entropy gain is obtained.

Finally, *to test whether a partial solution has solved the BDP problem*, refer to Theorem. 4.

Theorem 4. *For a BDP problem P, a solution S* ***solves*** *P iff each* ***AG*** *in S* ***meets the BDP requirement****: i.e. all TPs in it are covered and from one SIPG.*

Proof. Each AG is assigned a distinct AC. If all TPs in it are from one SIPG, they must be differentiated with the TPs in other SIPGs. Also, non-zero ACs are assigned as all TPs are covered. By Definition 2, S is feasible to P. □

E. Compact Solution Storage

For large scale BDP solutions, area codes can have *low weight but long length* since often a small subset of the deployed beacons cover some TP(s) together, due to *limited radiation range* of beacons and the constraints imposed by reducing beacon consumption.

Based on the above observation, to save space and speed up decoding: *1) ACTs can be stored like sparse matrices*; 2)an AC is split into *encoded bit fields*, each maps to *a group of beacons that never cover any TP simultaneously* (i.e. those beacons' bit positions in ACs are never set together). By information theory, a such group of n beacons can be encoded by $\lceil log_2(n+1) \rceil$ bits, assuming an all-zero bit field represent no beacon in group is used and another n codes represent a single beacon in group is used respectively. Clearly, bit length of a bit field in $\mathcal{A}$ is reduced from n to $O(log_2(n))$ via coding.

For example, consider the naive solution with n beacons adopted by iBeacon, where each TP is assigned a beacon. Since the deployed n beacons will not cover any TP together, they can be encoded by $\lceil log_2(n+1) \rceil$ bits. In other words, the ACT can be compressed to $\lceil log_2(n+1) \rceil$ columns.

In general, the deployed beacons can be coded in groups. As shown in Alg. 1, a compressed ACT can be obtained by firstly *creating an undirected graph G* **(lines 2-5)**. Each deployed beacon is a node while an edge between two nodes is created when they never both cover any TP (i.e. ANDing of these two beacon codes yields a zero-weight code).

Secondly, the *compressed area code format* in $\tilde{\mathcal{A}}$ can be constructed *via clique searching* in G. A clique of size n in G maps to n beacons that can be encoded in a bit field, since any TP can be covered by at most one of them. While G is non-empty, a max-size clique c with size greater than 2 is searched **(lines 6-7)**. If c exists, the beacons in c maps to a new bit field in $\tilde{\mathcal{A}}$, and all nodes in c with their associated edges are removed from G **(lines 8-9)**; otherwise, as encoding cannot shorten bit length of ACs, each remaining beacon is simply assigned a bit position **(lines 10-11)**.

Finally, each area code in $\mathcal{A}$ are *encoded using the compressed AC format* and sparse matrix representation is used if necessary **(line 12)**.

```
Input: ACT A
Result: compressed ACT: Ã
1  begin
2      add a node for each bcn in A to graph G
3      foreach bcn pair: b_i,b_j from A do
4          if ANDing b_i, b_j's bcn codes is all-zero then
5              add an edge (b_i, b_j) to G
6      while G ≠ ∅ do
7          if found a max-size clique c in G with size> 2 then
8              create a bit-field for c in Ã
9              remove nodes in c and any assoicated edges
                 from G
10         else
11             assign one bit to each remaining beacons in Ã
12     transform all area codes in A to Ã's area code format,
         store Ã as sparce matrix if necessary
```

Algorithm 1: cmrACT (cACT)

IV. Efficient Beacon Deployment

A. Pre-processing: power-level reduction

The *power-level reduction algorithm* proposed in [17] can be applied to remove redundant power-levels before running the proposed heuristic algorithm.

Observe that a group of power levels are *equivalent* for a BDP problem if their radiation patterns cover the same set of TPs. Therefore, computing the set of covered TPs for each power-level and keeping the min power-level for equivalent ones would not compromise solutions' beacon consumption. For example, consider the 8 power-levels in Fig. 1a, if their radiation patterns are specified by Fig. 1b, 4 power-levels: 0, 4, 6 and 7 are kept, which covers 1, 9, 21 and 25 TPs respectively; if the patterns are specified by Fig. 1f, 5 power-levels: 0 and $4-7$ are kept.

B. Heuristic Algorithm for Beacon Deployment

Algorithm 2 shows the main procedure RIT for generating a feasible BDP solution. Initially the AG list $aglst$ only includes

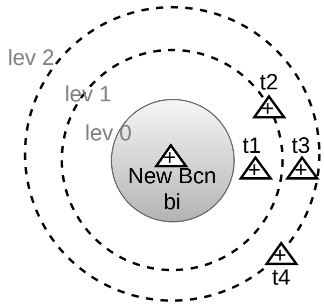


Fig. 3: power-level conflict demo

ag_0, a pseudo AG with all uncovered TPs **(line** 2**)**. During each iteration, $aglst$ always keeps all unidentified AGs and it is finally empty when a partial solution has met the BDP requirements.

While *keeping the partial solution "guaranteed feasbile"*(See Theorem 1), *the heuristic algorithm greedily adds new beacons which can gain most entropy then cover most uncovered TPs*.

A ***power-level conflict*** occurs if some power-level settings of a new beacon make current partial solution not "guaranteed feasible". To avoid these conflicts, *the algorithm will limit the radiation range of newly deployed beacons when necessary*. Take Fig. 3 for example, we try to install a new beacon at a TP b_i and it currently shares the same AC with the TPs t_1 to t_4 with installed beacons. Among the three available power-levels only $lev0$ is viable. To see this, $lev1$ makes b_i, t_1 and t_2 rely on extra beacons not on themselves to differentiate. Similarly, $lev2$ further makes b_i, t_3 and t_4 rely on others.

In implementation, each CP is assigned a *conflict counter*, initialized to 0 **(lines** $3-4$**)** and increments by 1 if current beacon generation fails due to conflicts. The beacon generation for a CP voluntarily give up for at most cft_{max} times when conflicts occur. Specifically, when $cft_{max} = 0$, the algorithm immediately generates a beacon at the chosen CP regardless of it has conflicts or not; in contrary, when cft_{max} is large, the algorithm favors deploying beacons with no conflicts.

As long as current partial solution cannot meet the BDP requirement **(line** 5**)**, procedure GAB is called to generate a beacon such that ambiguity of current partial solution can be reduced. RIT will be called N iterations to generate N solutions and pick the one with least beacon consumption.

```
   Input: BDP Problem
   Result: BDP Solution
1  begin
2      aglst ← {ag_0}
3      foreach CP i do
4          cft[i] ← 0
5      while aglst! = ∅ do
6          genAnyBcn()
```

Algorithm 2: RunIter (RIT)

GAB shown in Alg. 3 randomly picks an unused CP $bloc$ and either 1) generate a beacon to reduce ambiguity or 2) give up current trial if GAB decides to postpone the beacon generation for $bloc$ (judged by its conflict counter).

Firstly, GAB picks an unused CP $bloc$ at random and initialize $waitnext$ as $false$ **(lines** $2-3$**)**. In case $bloc$ is an identified TP or it is not a TP **(lines** $4-5$**)**, there is no limit on the power-level settings for $bloc$. Otherwise **(line** 6**)**, $bloc$ can conflict with any other unidentified CPs in the same AG. To prevent from conflicts, procedure $getPwrLim$ (gPLim) is called to compute the least upper-bound radiation range for $bloc$ **(line** 7**)**. If $pwrlim \neq \infty$ indicating conflicts, and conflict counter $cft[bloc] < cft_{max}$ **(line** 8**)**, then $cft[bloc]$ is incremented and $waitnext \leftarrow true$ to give up current trial. When $waitnext$ is $false$ **(line** 11**)**, every power-level with radiation range less than $pwrlim$ is tried and the one with best entropy gain **(line** 12**)** is chosen and saved **(line** 13**)**. *The motivation behind using $waitnext$ is that when conflicts occur it is sometimes better to choose another CP with less conflicts.*

```
    Input: partial BDP solution
    Result: a beacon with proper settings cbcn
1   begin
2       bloc ← randomly get an unused CP
3       waitnext ← false
4       if bloc is an identified TP or bloc is not a TP then
5           pwrlim = ∞
6       else
7           pwrlim ← getPwrLim(bloc)
8           if pwrlim ≠ ∞ and cft[bloc] < cft_max then
9               waitnext ← true
10              cft[bloc]++
11      if waitnext == false then
12          cbcn ← try each power-level with radiation range
              less than pwrlim at bloc, choose the one with best
              entropy gain
13          saveBcn(cbcn)
```

Algorithm 3: genAnyBcn (GAB)

Procedure gPLim in Alg. 4 returns the radiation range upper-bound $radlim$ at CP $bloc$, indicating $bloc$ should choose a power-level with radiation range less than $radlim$. Firstly all beacons within R_{max} are collected using circular range search algorithm. Only the beacons covering $bloc$ and whose positions are unidentified TPs in different SIPGs are kept. For each beacon i in $bset$, if i and $bloc$ are in the same SIPG, $radlim$ will be set to i's radiation range $i.radrng$ if $i.radrng < radlim$.

```
   Input: CP bloc
   Result: limit of radiation range
1  begin
2      radlim ← ∞
3      bset ← beacons within R_max which (1)cover bloc and
         (2) whose postions are unidentified TPs not in bloc's
         SIPG
4      baid ← gAreaCode(bloc)
5      foreach i ∈ bset do
6          if gAreaCode(i) == baid then
7              radlim ← min(radlim, i.radrng)
8      return radlim
```

Algorithm 4: getPwrLim (gPLim)

Alg. 5 shows the procedure for getting the area code at loc. Firstly, the returned area code aid is cleared and all beacons within R_{max} are collected by a 2D range search algorithm.

Then, all beacons covering $bloc$ sets their corresponding bits in aid [1].

```
Input: TP loc
Result: area code of loc
begin
    intitialize area code aid to 0
    bset ← range search beacons within R_max
    foreach i ∈ bset do
        if i covers loc then
            aid[i] = 1
    return aid;
```

Algorithm 5: gAreaCode (gAC)

Procedure $saveBcn$ in Alg. 6 integrates the beacon candidate $bbcn$ into current partial solution and adjust the AGs and their properties accordingly. Firstly, TPs covered by $bbcn$ are grouped by their previous area codes into a AG list ags **(line** 2). Each AG in ags is further split into two AGs, ag^1 and ag^0, containing the TPs covered by $bbcn$ or not accordingly. Finally, depending on whether ag^0 or ag^1 meets the BDP requirement, it should be removed from or appended to the ambiguous group list $aglst$ **(line** 5**-**8).

```
  Input: new beacon bbcn with settings
  Result:
1 begin
2     ags ← group TPs covered by bbcn according to their
        previous area codes
3     foreach ag ∈ ags do
4         split ag into ag^1 and ag^0, containing TPs covered
            by bbcn or not respectively
5         if ag^0 meets BDP requirement then
6             remove ag^0 from aglst
7         if ag^1 cannot meet BDP requirement then
8             append ag^1 to aglst
```

Algorithm 6: saveBcn (sBcn)

Theorem 5. ***Correctness of the heuristic algorithm****: the proposed heuristic algorithm in Alg. 2 is applicable to arbitrary radiation patterns and converges.*

Proof. 1) **Applicability**: no procedures called by Alg. 2 are dependent on the shape of radiation patterns. Thus, the algorithm is applicable to any patterns.
2) **Correctness**: Alg. 2 keeps on generating new beacons by calling GAB (See Alg. 3) until BDP is achieved. Each CP is considered for at most cft_{max} times in Alg. 2. In the end, a CP is either a) *ditched* if no power-level settings of it can improve the solution's entropy or current solution has already met the BDP requirements or b) *chosen* if it can improve the entropy of current partial solution and be saved. Thus, after at most cft_{max} consecutive GAB calls, the entropy of current partial solution must increase. Thereby, the algorithm converges eventually when BDP is achieved. □

[1]To support arbitrary radiation patterns, a geometry software library capable of testing whether a given coordinate is inside a shape is required. To our knowledge, CGAL supports both circular and polygon radiation patterns while non-polygon ones can be approximated by polygons.

C. Post-processing: shrinking beacon consumption

The best solution acquired by Alg. 2 can still contain redundant beacons as beacons are added sequentially to a solution, thereby some beacons' contribution to remove ambiguity can probably be achieved by some newly added beacons during solution generation. This phenomenon is more significant when the problem scale is large.

To address the above issue, Alg. 7 tries to delete useless beacons from a solution $csol$ one by one. Note that *Alg. 7 supports dropping the beacon consumption of any sub-optimal BDP solutions generated by any beacon deployment algorithm*, not limited to our ILP or heuristic algorithm.

Initially all beacons in $csol$ are shuffled randomly into $bset$ and all used ACs are stored in $ACset$ **(lines 1-3)**. The algorithm then checks whether each beacon i in $bset$ can be removed sequentially **(line 4)**. Suppose the i_{th} beacon in $csol$ use the i_{th} bit, its removal is equivalent to reset the i_{th} bit column in ACT and affect all ACs whose i_{th} bits are set originally. Thus, the affected ACs should be updated in ACT if beacon i can be removed.

Let all ACs affected by removal of i be stored in $affACs$ **(line 5)**. For each area code $ac \in ACset$ whose i_{th} bit is 1, removal of i produces a novel AC: tac by clearing ac's i_{th} bit **(lines 7-9)**. A removal fails if 1) tac is zero, which means a TP becomes uncovered or 2) tac and ac both appeared in $ACset$ but belonging to different SIPGs, which means two TPs become unidentified **(lines 10-12)**. Otherwise, the removal succeeds for ac, and all affected ACs should be recorded **(lines 13-14)**. Note that beacon i should be kept if any AC in $ACset$ fails the above tests after ditching i. Otherwise, i could be safely removed from $csol$ **(lines 4-14)**.

If removal of i succeeds, each affected ACs should replace its corresponding old AC in $ACset$. Beacon i then can be safely removed from $csol$ **(lines 15-19)** and the algorithm proceeds to test the next beacon.

```
   Input: beacon solution csol
   Result: shrinked csol
1  begin
2      bset ← randomly shuffled beacons in csol
3      ACset ← used area codes in csol
4      foreach i ∈ bset do
5          affACs ← ∅
6          suc ← true
7          foreach ac ∈ ACset do
8              if ac[i] == 1 then
9                  tac ← ac,tac[i] = 0
10                 if (tac == 0) or ( tac ∈ ACset and
                     (tac,ac) ∉ same SIPG) then
11                     suc ← false
12                     break
13                 else
14                     insert ac into affACs
15         if suc == true then
16             foreach aac ∈ affACs do
17                 tac ← aac,tac[i] = 0
18                 ACset = (ACset \ aac) ∪ tac
19             remove i from csol
20     return csol
```

Algorithm 7: shrkSol (skSl)

V. Simulation Results

All heuristic simulations are for BDUP problems using Estimote beacon's power-level settings, conducted on a PC with dual 2.2GHz Intel CPU cores and 4G RAM; the ILP results are quoted from [9], whose simulations are run on a server with quad 3.6GHz Intel CPU cores and 16G RAM. *Note that the heuristic algorithm is applicable to any radiation patterns as proved in Theorem 5, but we only implemented the circular power-level version in C++ currently.*

For Table I and Fig. 4, the AOI is $60m$ by $60m$, attenuation factor α varies from 3 to 5 for the log radio propagation model. The **"save"** metric in Table I is defined as the mean number of beacons per TP.

Table I compares the performance of the proposed heuristic algorithm and the ILP in [9] in beacon consumption and running time. The heuristic algorithm runs for 100 iterations with cft_{max} set to 2; the solution with lowest beacon consumption is obtained by Alg. 2 and post-processed by Alg. 7.

Fig. 4a, 4c and 4e demonstrate the best heuristic solutions for $|\mathbb{T}| = 15 * 15$ when $\alpha = 3$,4 and 5. By contrast, Fig. 4b, 4d and 4f illustrate the solutions for $|\mathbb{T}| = 12*12$ when $\alpha = 3$ to 5 respectively.

Fig. 4g compares the "save" metric achieved by ILP and the proposed heuristic algorithm, when $|\mathbb{T}|$ is 10*10, 12*12, 15*15 and 20*20 respectively and α varies from 3 to 5. Clearly, the "save" metric increases with the growth of $|\mathbb{T}|$ under all settings for both methods. The *heuristic algorithm consumes about* 1.14 *to* 1.67 *times of beacons compared to the ILP results, while the running time of ILP simulations is several magnitudes longer*, as shown by Fig. 4h. In addition, the *running time grows almost linearly for the proposed algorithm when* $|\mathbb{T}|$ *changes.*

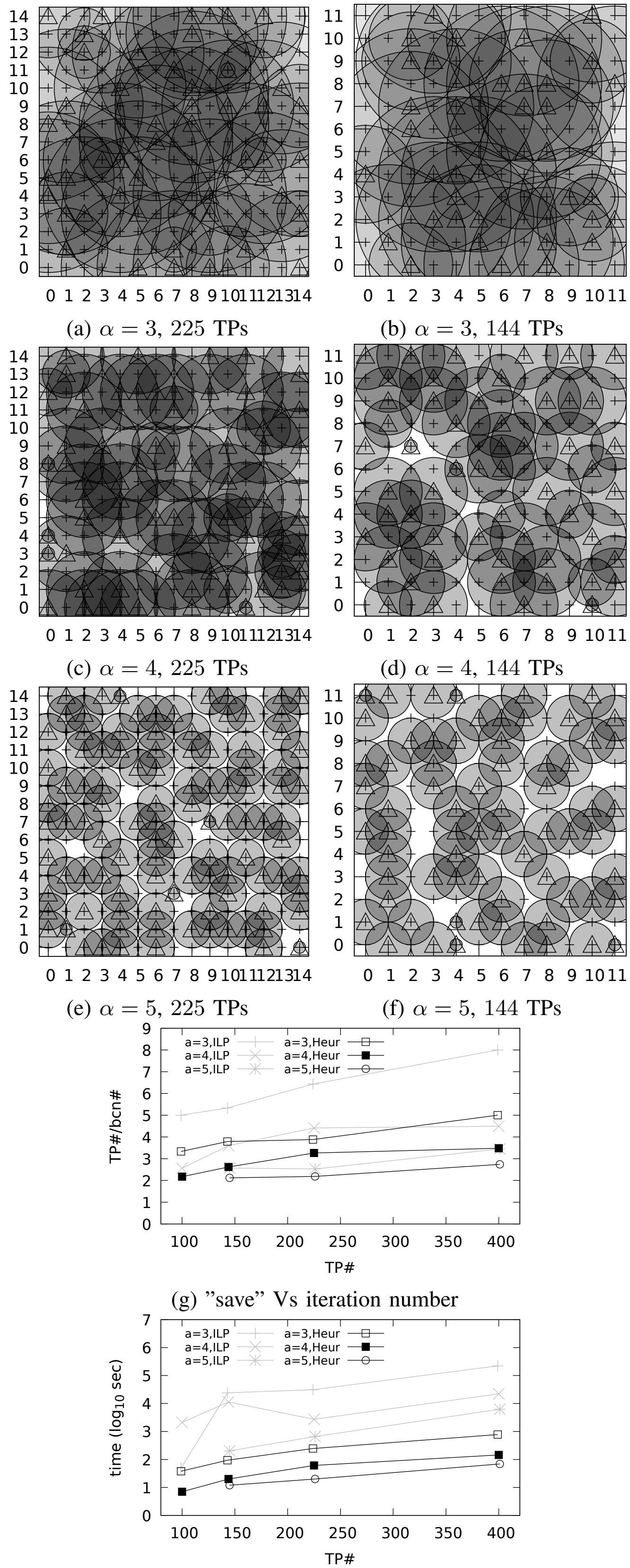
TABLE I: ILP vs. Heuristic details: 60 by 60 meters, $\theta = -97$dBm.

Input				ILP				Heur 100its		
α $\vert\mathbb{T}\vert$	ϕ	R_{max}	$\vert\mathbb{V}\vert$	bcn#	save	time(s)	ILPgap	bcn#	save	time(s)
3 400	101	$5.7g$	7	50	8.0	221568	40.4%	80	5.0	777
3 225	61	$4.3g$	6	35	6.4	31244	31.4%	58	3.9	246
3 144	37	$3.4g$	6	27	5.3	24138	23.6%	38	3.8	94
3 100	25	$2.9g$	4	20	5.0	49	0.0%	30	3.3	38
4 400	21	$2.5g$	4	89	4.5	22199	29.9%	115	3.5	145
4 225	13	$2.1g$	4	51	4.4	2723	24.0%	69	3.3	61
4 144	9	$1.5g$	3	40	3.6	11336	11.0%	55	2.6	20
4 100	5	$1.2g$	2	39	2.6	2115	19.0%	46	2.2	7
5 400	9	$1.7g$	3	116	3.4	6252	25.4%	146	2.7	69
5 225	5	$1.0g$	2	89	2.5	659	27.0%	103	2.2	20
5 144	5	$1.0g$	2	56	2.6	203	22.9%	68	2.1	12

Fig. 5 tries to find the optimal value for setting maximum iterations N and maximum tolerable conflicts cft_{max}; and investigates the performance when $|\mathbb{T}|$ varies and g stays constant.

As Fig. 5a shows, *best "save" is achieved when cft_{max} is* 2 *or* 4 *in most cases*, while higher cft_{max} doesn't lead to much improvement. As plotted in Fig. 5b, the average running time per iteration is stably less than 2 seconds for most cases when cft_{max} changes, and hits about 9 seconds when the problem scale is large, i.e. $\alpha = 3$ and $|\mathbb{T}| = 20 * 20$.

Fig. 5c-5d examine the performance when problem scale changes: i.e. $|\mathbb{T}|$ is $20 * 20$, $40 * 40$, $60 * 60$ and $80 * 80$,and



(a) $\alpha = 3$, 225 TPs (b) $\alpha = 3$, 144 TPs

(c) $\alpha = 4$, 225 TPs (d) $\alpha = 4$, 144 TPs

(e) $\alpha = 5$, 225 TPs (f) $\alpha = 5$, 144 TPs

(g) "save" Vs iteration number

(h) Heuristic run time

Fig. 4: Beacon Deployment Heur Vs. ILP: $60 \cdot 60m^2$ meters, $\theta = -97$dBm

grid gap g is 3m and 5m respectively. *With the growth of* $|\mathbb{T}|$ as Fig. 5c shows, *a higher "save" value can be achieved. By applying Alg. 7, "save" value can be further improved.* The running time for the proposed algorithm also grows linearly with respect to $|\mathbb{T}|$.

Fig. 5f shows the running time of Alg. 7 when $|\mathbb{T}|$ changes. Clearly the *shrinking algorithm scales well with respect to* $|\mathbb{T}|$ and the maximum running time per iteration in simulations is about 5 seconds.

Finally, Fig. 5e studies the "save" value when the maximum iteration number changes. According to the simulation results, *the algorithm converges quickly and hits sub-optimal at about* 100 *iterations in most cases.*

VI. Conclusion

In this paper, we propose an entropy-based heuristic algorithm for solving large-scale BDP problem incorporating with a coding method in order to reduce memory consumption of solutions. A novel post-processing beacon shrinking algorithm is introduced that is applicable to any beacon deployment algorithm. Based on the proposed approach, the beacon consumption is 1.14 to 1.67 times higher than that by the optimal ILP, while achieving much shorter running time and smaller memory consumtion in all the simulated scenarios.

References

[1] A. Mackey, P. Spachos, and K. Plataniotis, "Smart parking system based on bluetooth low energy beacons with particle filtering," *IEEE Systems Journal*, vol. PP, pp. 1–12, 02 2020.

[2] D. Lin and Z. Mi, "Electric fence planning for dockless bike-sharing services," *Journal of Cleaner Production*, vol. 206, 09 2018.

[3] W. He and P.-H. Ho, "On achieving cyber-physical real-time snapshot acquisition in billboard/signage networks," *IEEE Internet of Things Journal*, vol. 3, pp. 1–1, 12 2016.

[4] A. Ito, Y. Hiramatsu, H. Hatano, M. Sato, M. Fujii, Y. Watanabe, F. Sato, and A. Sasaki, "Navigation system for sightseeing using ble beacons in a historic area," in *2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2016, pp. 171–176.

[5] K. Shuang, X. Ren, Q. Yang, R. Li, and J. Loo, "Aela-dlstms: Attention-enabled and location-aware double lstms for aspect-level sentiment classification," *Neurocomputing*, vol. 334, pp. 25 – 34, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231218315054

[6] C.-K. Ke, M. Wu, Y. Chan, and K. Lu, "Developing a ble beacon-based location system using location fingerprint positioning for smart home power management," *Energies*, vol. 11, p. 3464, 12 2018.

[7] P. Spachos and K. Plataniotis, "Ble beacons in the smart city: Applications, challenges, and research opportunities," *IEEE Internet of Things Magazine*, vol. 3, pp. 14–18, 03 2020.

[8] M. Gao, X. Zhang, T. Zhang, C. Chen, Z. Wang, Z. Lu, W. Ding, and J. Ouyang, "An experimental study on wechat-based large scale indoor localization system," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, 2018, pp. 330–338.

[9] W. He, P.-H. Ho, and J. Tapolcai, "Beacon deployment for unambiguous positioning," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1370–1379, 10 2017.

[10] "Estimote power levels." [Online]. Available: https://forums.estimote.com/t/measured-power-values/2977

[11] A. Mackey, P. Spachos, L. Song, and K. Plataniotis, "Improving ble beacon proximity estimation accuracy through bayesian filtering," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 01 2020.

[12] Blasio, de, Rodríguez-Rodríguez, García, Carmelo, and Quesada-Arencibia, Alexis, "Beacon-related param of bluetooth low energy: Development of a semi-automatic system to study their impact on indoor positioning systems," *Sensors*, vol. 19, p. 3087, 07 2019.

[13] G. Piccinni, G. Avitabile, G. Coviello, and C. Talarico, "Modeling of a re-configurable indoor positioning system based on software defined radio architecture," in *2018 New Generation of CAS (NGCAS)*, 11 2018, pp. 174–177.

[14] H. Cheng, H. Liu, W. Wang, and J. Li, "A compact and cost-effective ble beacon with multiprotocol and dynamic content advertising for iot application," *IEEE Internet of Things Journal*, vol. 7, pp. 2309–2320, 03 2020.

[15] C. Han, A. Misra, and S. Cheng, "Mobility-driven ble transmit-power adaptation for participatory data muling," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, 12 2018, pp. 962–971.

[16] S. S. Chawathe, "Beacon placement for indoor localization using bluetooth," in *Int. IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct 2008, pp. 980–985.

[17] W. He, D. Wang, Y. Xiong, L. Chen, and Y. Liu, "On bounded positioning via bluetooth," *Journal of Technology*, in Chinese, under review.

[18] R. Luo and T.-J. Hsiao, "Indoor localization system based on hybrid wi-fi/ble and hierarchical topological fingerprinting approach," *IEEE Transactions on Vehicular Technology*, vol. PP, pp. 1–1, 09 2019.

[19] S. Safavi, U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed localization: A linear theory," *Proceedings of the IEEE*, vol. 106, no. 7, pp. 1204–1223, 7 2018.

[20] R. M. Buehrer, H. Wymeersch, and R. M. Vaghefi, "Collaborative sensor network localization: Algorithms and practical issues," *Proceedings of the IEEE*, vol. 106, no. 6, pp. 1089–1114, 6 2018.

[21] J. B. Andersen, T. S. Rappaport, and S. Yoshida, "Propagation measurements and models for wireless communications channels," *IEEE Communications Magazine*, vol. 33, no. 1, pp. 42–49, Jan 1995.

[22] E. A. Doumith, S. A. Zahr, and M. Gagnaire, "Monitoring-tree: An innovative technique for failure localization in WDM translucent networks," in *GLOBECOM*, 2010, pp. 1–6.

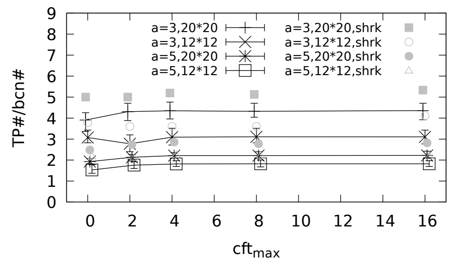
[23] W. He, P.-H. Ho, B. Wu, and J. Tapolcai, "On identifying SRLG failures in all-optical networks," *Optical Switching and Networking*, vol. 10, no. 1, pp. 77 – 88, 2013, advances in Optical Networks Control and Management. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1573427712000641

Wei He Dr. Wei He received his B.Sc. degree in computer science from Fudan University and M.SE. degree from Peking University, in 2005 and 2008 respectively; and Ph.D. degree from School of Computer Science, University of Waterloo in 2013. He is now an assistant professor in the School of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai, China. His current research interests include high performance wireless positioning, Internet of Things, survivable network design, and Cyber-physical systems.

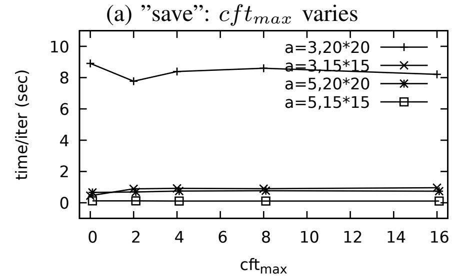
Pin-Han Ho Dr. Pin-Han Ho received his B.Sc. and M.Sc. degree from the Electrical Engineering dept. in National Taiwan University in 1993 and 1995, and Ph.D. degree from Queen's University at Kingston at 2002. He is now a professor in the department of Electrical & Computer Engineering, University of Waterloo, Canada. His current research interests include survivable network design, Internet of Things etc..

Dong Wang Dr. Dong Wang received the Ph.D. degree in computer science from Tongji University in 2013. He works at Shanghai Institute of Technology, Shanghai, China, where he works on intelligent decision support systems, information services, trusted computing, and social networks. His research centers around machine learning from relational knowledge representations and graph-structured data as well as its applications in artificial intelligence and cognitive science.

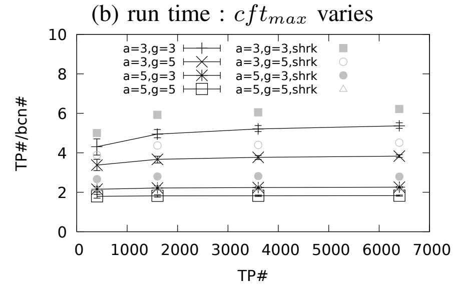
Lizhong Xiao Dr. Lizhong Xiao received his doctroal degree from East China University of Science and Technology in 2007. He is now an associate professor in the School of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai, China. His research interest includes software engineering, data mining and Intelligent Manufacturing.



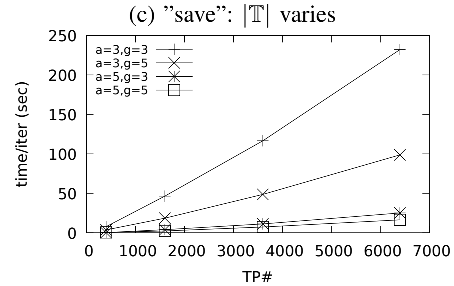
(a) ”save”: cft_{max} varies



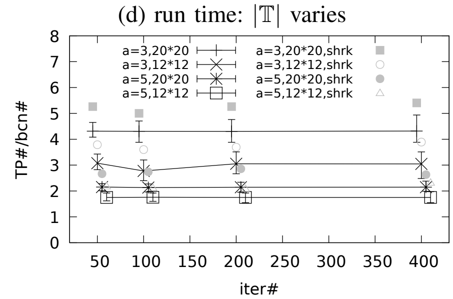
(b) run time : cft_{max} varies



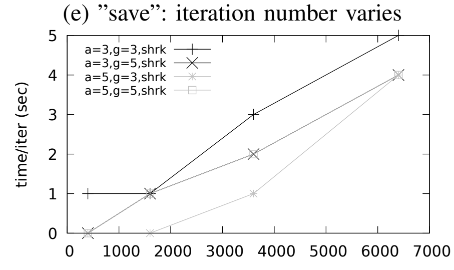
(c) ”save”: $|\mathbb{T}|$ varies



(d) run time: $|\mathbb{T}|$ varies



(e) ”save”: iteration number varies



(f) shrinking time : $|\mathbb{T}|$ varies

Fig. 5: Heuristic parameter tuning