An Approximate Solution of the Multiple Watchman Routes Problem with Restricted Visibility Range

Jan Faigl

***Abstract*—In this paper, a new self-organizing map (SOM) based adaptation procedure is proposed to address the Multiple Watchmen Route Problem with the restricted visibility range in the polygonal domain $\mathcal{W}$. A watchman route is represented by a ring of connected neurons weights that evolves in $\mathcal{W}$ while obstacles are considered by approximation of the shortest path. The adaptation procedure considers a coverage of $\mathcal{W}$ by the ring in order to attract nodes towards uncovered parts of $\mathcal{W}$. The proposed procedure is experimentally verified in a set of environments and several visibility ranges. Performance of the procedure is compared with the decoupled approach based on solutions of the Art Gallery Problem and the consecutive Traveling Salesman Problem. The experimental results show suitability of the proposed procedure based on relatively simple supporting geometrical structures enabling application of SOM principles to watchman route problems in $\mathcal{W}$.**

***Index Terms*—Watchman route problem, Self-organizing maps**

I. Introduction

Self-Organizing Maps (SOM) have been already applied to various combinatorial routing problems [32], [14], [23], [34]. The most representative problem is the Traveling Salesman Problem (TSP), which deals with finding a shortest tour that visits given set of cities. The TSP has been addressed by several SOM algorithms, e.g. [33], [9], [6], [10]. However, these approaches are mainly focused on the Euclidean TSP in which distances between neurons and cities can be efficiently computed as the Euclidean distance, while non-Euclidean problems are out of the community interest, probably due to difficulty of distance determination [27]. On the other side, techniques from the operational research provide solutions for the non-Euclidean TSP, where distances between cities are precomputed and stored in the so-called distance matrix.

The *Watchmen Route Problem* (WRP) [8] is a routing problem in which points that have to be visited are not explicitly prescribed. The problem is to find a closed shortest path in a polygon P such that all points of P are visible from at least one point at the path. Shermer called this problem *hybrid visibility problem* [30] because beside the visibility concept (the visibility graph), other geometric or conceptual properties are involved in the problem. The difficulty of the problem is that distances between points to be visited cannot be precomputed, because the points are not known or the number of possible positions is too high. Therefore a simplification of the problem or approximations should be considered. In [5], Aras et at. note that neurons of SOM tend to learn the properties of the underlying distribution of the space in which they operate. Such a feature can be advantageous in the hybrid visibility problems, however SOM has not yet been applied to this type of problems.

The WRP is NP-hard for a polygon with holes, and the problem is studied for restricted classes of polygons, for which approximate and optimal algorithms have been found [20], [16]. The problem can be considered as a variant of the well known *Art Gallery Problem* (AGP) posed by Klee in 1973. Watchmen (guards) are static in the AGP, and the problem is to minimize the number of guards. Once the guards are found, a watchman route can be found as a solution of the TSP where guards denote cities.

The *Multiple Watchman Routes Problem* (MWRP) deals with several watchmen and can be formulated for two criteria: the minimization of the total travelled path (MinSum) and the minimization of the longest route (MinMax). Both variants are known to be NP-hard for a polygon with holes. Probably the first heuristic approach for the MWRP in the polygonal domain has been presented in [26]. The algorithm is based on a set of static guards that are used to determine the minimum spanning tree from the pairwise shortest paths between guards. The tree is split to construct routes that are shortened by vertex substitutions and removing of redundant vertices. The heuristic algorithm is able to address both criteria (MinSum and MinMax). Even though the algorithm has been used in several problems, only an unrestricted visibility has been considered, which limits the real applicability of the algorithm.

Capabilities of real sensors, like cameras or range finders, are limited in resolution, sensing range or frequency, therefore the problem to "see" an object or workspace is studied for various sensing constraints [28], [36]. If the visibility range is restricted to a distance d, two variants of the WRP can be found in the literature [35]. The *d-watchman route problem* is a variant to see only the boundary of the polygon, while the *d-sweeper route problem* is to sweep a polygonal floor using a circular broom of radius d, so that the total travel of the broom is minimized [24]. In these problems, the notion of *d-visibility* is considered, i.e. two points p and q in a polygon P are called *d-visible*, if the line segment joining them is contained in P and the segment length is less or equal to d.

The MWRP with d-visibility is addressed in this paper. Particularly the problem is formulated as follows: *for a given polygon $\mathcal{W}$ (possibly with holes) find k closed paths in $\mathcal{W}$ considering MinMax criterion such that all points of $\mathcal{W}$ are*

This work has been supported by the Ministry of Education of the Czech Republic under program "National research program II" by the project 2C06005.

J. Faigl is with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27, Prague 6, Czech Republic (e-mail: xfaigl@labe.felk.cvut.cz).

d-visible from at least one point at the paths. The proposed algorithm uses Kohonen's self-organizing map and provides approximate solution of the problem. To our best knowledge the proposed algorithm is the first application of SOM to the WRP and it is probably the first soft-computing approach to address the MWRP with d-visibility in the polygonal domain. The algorithm is based on the adaptation schema for the Multiple Traveling Salesman Problem (MTSP) with the MinMax criterion [34], which has been applied only to the Euclidean TSP. Therefore to apply the schema in the polygonal domain it is necessary to consider a path among obstacles.

The rest of this paper is organized as follows. The next section presents used notation and terminology. Section III is dedicated to the related work in which selected reference algorithm is described in Section III-A, the used SOM adaptation procedure in Section III-B and its extension to the polygonal domain in Section III-C. The main contribution of this paper, an adaptation procedure for the WRP with d-visibility, is presented in Section IV. Its MWRP variant is described in Section V. Experimental results verifying the proposed WRP and MWRP algorithms are presented in Section VI together with a comparison with the reference (AGP+TSP) approach. A discussion of the proposed algorithms and further possible improvements are presented in Section VII. Conclusions are summarized in Section VIII.

II. Used Terms and Notation

SOM is considered in the polygonal domain $\mathcal{W}$, therefore few terminology notes are presented in this section to clarify used terms and symbols for the supporting geometrical structures.

A world is represented by a polygonal map $\mathcal{W}$ consisting of $N_{\mathcal{W}}$ vertices. $\mathcal{W}$ is a closed, multiply connected region, whose boundary is a union of $N_{\mathcal{W}}$ line segments, forming $h+1$ closed polygonal cycles, where h is the number of holes (obstacles). A distance between two points inside $\mathcal{W}$ is a length of a path among obstacles that can be a straight line segment or consists of vertices. Thus, a path between two points s and t consists of a finite number of line segments joining the points and vertices.

$\mathcal{W}$ can be divided into a set of non-overlapping convex polygons that are formed from vertices. Such convex polygons are called cells and represent *convex polygon partition* of $\mathcal{W}$, i.e. each cell C forms a closed polygonal cycle of line segments joining vertices. A line segment is called *diagonal* if it connects two nonadjacent vertices and it is contained in $\mathcal{W}$. A point inside $\mathcal{W}$ is always inside some cell and a path between two points $s \in C_s$ and $t \in C_t$ can be constructed from the shortest path between vertices of C_s and C_t. Weights of the ith neuron represent a point ν_i (called node) that lies in $\mathcal{W}$, therefore ν_i is always inside some cell. Such a cell containing the node ν is denoted as C_ν.

An additional used supporting division of $\mathcal{W}$ is a *triangular mesh* that is formed from the vertices of $\mathcal{W}$ and additional points placed inside $\mathcal{W}$. More formally, a triangular mesh $\mathcal{T}$ is a triplet $\mathcal{T} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{T})$, where $\boldsymbol{V}$ is a set of mesh vertices, $\boldsymbol{E}$ is a set of edges $e \in \boldsymbol{E}$, $e = (v_i, v_j), v_i, v_j \in \boldsymbol{V}, v_i \neq v_j$, $\boldsymbol{T}$ is a set of triangles $T \in \boldsymbol{T}$, $T = (\{v_i, v_j, v_k\}, \{e_i, e_j, e_k\})$ where $v_i, v_j, v_k \in \boldsymbol{V}$, $e_i, e_j, e_k \in \boldsymbol{E}$, $e_i = (v_k, v_i)$, $e_j = (v_i, v_j)$, $e_k = (v_j, v_k)$. Moreover, the mesh vertices are used to find a *convex cover set* of $\mathcal{W}$ that is a collection of convex sub-polygons of $\mathcal{W}$ whose union is exactly $\mathcal{W}$.

To avoid possible confusions, the term *mesh vertices* is used for the vertices of $\mathcal{T}$ and the term *map vertices* is used for the vertices of $\mathcal{W}$ in cases where it can be misinterpreted. In all other cases, the single word *vertices* represents the *map vertices*. The term *node* is strictly reserved to the values of weights of a neuron.

An overview of the used symbols is shown in Table I.

TABLE I: Used symbols

Symbol	Description
$\mathcal{W} \subset \mathbb{R}^2$	a polygonal domain representing the guarded world
$N_{\mathcal{W}}$	a number of vertices of $\mathcal{W}$
d	a visibility range of watchman or guard
n	a number of guards/cities in the related AGP/TSP
m	a number of neurons representing a route
k	a number of salesmen/watchmen in the MTSP/MWRP
$\|s, t\|$	the Euclidean distance between points s and t
$\|S(v_i, v_j)\|$	a length of the shortest path between two vertices of $\mathcal{W}$
$\boldsymbol{P}$	a set of convex polygons
v_i	a vertex of the polygonal domain $\mathcal{W}$
ν_i	a node representing weights of the ith neuron
$\mathcal{T}$	a triangular mesh of $\mathcal{W}$, $\mathcal{T} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{T})$
N_T, N_V	a number of triangles and a number of mesh vertices
N_C	a number of convex polygons of a cover set
G, μ, α	parameters of the used SOM adaptation schema

III. Related Work

A. Reference Algorithm

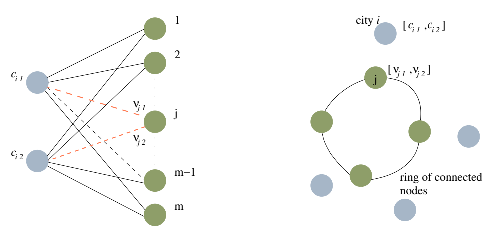
A decoupled approach (AGP+TSP) represents a feasible way how the WRP can be solved. The main advantage of the approach is ability to deal with d-visibility, because computationally feasible algorithm that will directly provides solutions of the WRP with d-visibility in the polygonal domain is not known. A sensor placement algorithm [19] is used to find a solution of the AGP. The algorithm provides guards with d-visibility and it is based on a deterministic decomposition of a polygon $\mathcal{W}$ into a set of convex polygons. Each convex polygon is guarded by one guard, and to satisfy $d-$visibility, a distance from the guard to a vertex of the guarded polygon has to be less than d. The primal convex partition is found by Seidel's algorithm [29] and if a convex polygon is too large, it is divided into convex sub-polygons, until each sub-polygon can be covered by one guard with d-visibility. The complexity of the algorithm is linear with the number of found guards [19].

A solution of the WRP is found as a solution of the TSP on a graph $G(V, E)$, where V denotes the found guards and E is a set of edges with costs derived from the length of the shortest path between guards. The paths are found by Dijkstra's algorithm in $O(nn_e \log(n + N_{\mathcal{W}}))$ on the visibility graph, which is found in $O((n + N_{\mathcal{W}})^2)$ [25], where $N_{\mathcal{W}}$ denotes the number of polygon vertices, n is the number of guards and n_e is the number of edges of the visibility graph. Without loss of generality $G(V, E)$ is assumed to be complete. The TSP can be solved exactly by the Concorde solver [3].

B. SOM Adaptation procedure for the TSP

The adaptation schema for the TSP [33] is used as the main adaptation schema for the new proposed algorithm for the WRP. A detailed overview of the schema is presented in the next paragraphs, because the schema is applied in the polygonal domain $\mathcal{W}$. The overview provides insight how the procedure relates with the used approximations and supporting geometric structures that enable application of SOM in $\mathcal{W}$. Thus, a reader can skip this part and continue the reading from Section III-C in which the used approximation of the shortest path is described.

The used two-layered competitive learning network consists of two dimensional input vectors and an array of output units. An input vector i represents coordinates (c_{i1}, c_{i2}) of the city c_i and weights ν_{j1}, ν_{j2} can be interpreted as coordinates of the node ν_j, see Fig. 1. Connected nodes form a ring that evolves in the problem domain according to the following self-organizing adaptation procedure.



(a) (b)

Fig. 1: Schema of the two-layered neural network and associated geometric representation.

The network is initialized with small random connection weights and cities are then sequentially applied to the network in a random order. For a given city, the output nodes compete to be the winner node according to their distance to the city. The weights of the winner node and its neighbouring nodes are updated in order to get closer to the presented city according to the neighbouring function f. The adaptation function moves a node towards the city c_i by the rule $\nu_j' = \nu_j + \mu f(G, l)(c_i - \nu_j)$, where μ is the fractional learning rate. The used neighbouring function is $f(G, l) = \exp(-l^2/G^2)$ for $l < 0.2m$ and $f(G, l) = 0$ otherwise, where G is the gain parameter, l is the distance (in the number of nodes) of a node from the winner measured along the ring, and m is the number of nodes in the ring that is set to $m = 2.5n$, where n is the number of cities. The gain G is decreased after each complete presentation of the cities to the network according to the gain decreasing rate α, i.e. $G = G(1-\alpha)$. Authors of [34] recommend to set the initial value of G according to the formula $G_0 = 0.06 + 12.41n$ and values of the learning and decreasing rates μ=0.6 and α=0.1. The presentation of all cities to the network is repeated until all cities have their winner nodes closer than given threshold.

An inhibition mechanism [33] guarantees that a distinct winner node is found for each city during one adaptation step. So, a city tour can be found by traversing the ring at the end of each adaptation step. The final length of the city tour is computed as the sum of city–city distances.

1) The SOM procedure for the MTSP-MinMax: In MTSP, the problem is to find a tour for each of k salesmen, thus an individual ring of nodes is created for each salesman. All tours start and finish at the same city that is called *depot*. The adaptation procedure must ensure that all tours are connected with the depot, therefore a winner node from each ring is selected and adapted to the depot at the beginning of each adaptation step. After that, other cities are presented to the network in a random order and a winner node is selected from all non-inhibited nodes like in the TSP variant. To address the MinMax criterion authors of [34] proposed a competitive rule $\nu^\star = \text{argmin}_\nu |c, \nu| \cdot (1 + (l_\nu - l_{avg})/l_{avg})$, where $|c, \nu|$ denotes the Euclidean distance between the city c and the node ν, l_ν is the length of the ring into which the node ν belongs and l_{avg} is the average length of the rings. The rule prefers nodes from shorter rings and it leads to minimize the longest tour. Same number of neurons is used in all rings and it is set to $m = 2.5n/k$.

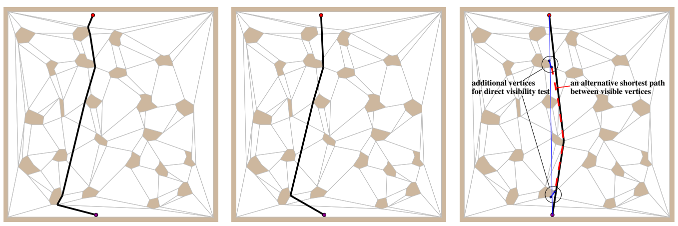
The SOM adaptation procedure relies on efficient determination of the winner node that uses a node–city distance. The adaptation of a node towards the presented city uses a node–city path along which the node is moved. Moreover, the shortest path between two nodes is needed to compute a length of the ring in the MTSP-MinMax. A suitable approximation of the shortest path in $\mathcal{W}$ is described in the next section.

C. Approximation of the Shortest Paths

The node–city and node–node distances (paths) are needed to use the SOM algorithm for the MTSP-MinMax in $\mathcal{W}$. The exact determination of the node–city distance and path can be supported by the precomputed *Shortest Path Map* (SPM), which is a planar division with respect to a point (city). The SPM provides the shortest distance to the point in $O(\log N_\mathcal{W})$ and the shortest path in $O(\log N_\mathcal{W} + l)$, where $N_\mathcal{W}$ is the number of vertices and l is the number of bends in the path [22]. The main drawback of the precomputed SPM is the required space, because the SPM has to be found for each city, which is impractical for hundreds or thousands of cities. Alternatively the path can be found by a construction of the SPM that can also be used for the two points (node–node) path, e.g. in $O(N_\mathcal{W} \log N_\mathcal{W})$ [17], or in $O(\log N_\mathcal{W})$ using $O(N_W^{11})$ space [7]. Even though the SPM provides exact shortest path, a simpler and computationally feasible approximation of the shortest path between two points in $\mathcal{W}$ seems to be sufficient for the SOM adaptation procedure [13]. The main idea of the approximation is based on precomputed shortest paths between map vertices and a supporting convex polygon partition. The all shortest paths can be found in $O(N_\mathcal{W}^3)$ and a convex partition can be found in $O(N_\mathcal{W} \log N_\mathcal{W})$ [29] with the required space $O(N_\mathcal{W}^2)$. A convex polygon partition $\boldsymbol{P}$ is a set of convex polygons C_i, $\boldsymbol{P} = \{C_1, C_2, \ldots, C_l\}$. A path between two points is found as follows.

Let two points p_1 and p_2 be inside particular cells C_1 and C_2. A path between the points is constructed from the shortest path between vertices of each cells $S(v_1, v_2)$, where $v_1 \in C_1$ and $v_2 \in C_2$. Vertices v_1, v_2 are selected to minimize the path length: $|p_1, v_1| + |S(v_1, v_2)| + |v_2, p_2|$, where $|.,.|$ denotes the

Euclidean distance of two points and $|S(.,.)|$ is a length of the shortest path between two vertices. An example of the path over cell vertices v_1 and v_2 is shown in Fig. 2a. A path is a sequence of vertices $(v_1, \ldots, v_2)$ and can be refined by a direct visibility test between the points and particular vertex v_i of the path, see Fig. 2b. The visibility test is similar to the method described in [18], instead of a triangulation the convex partition is used. The path approximation can be improved by an additional visibility test of vertices of obstacle edges that intersect the segment (p_1, p_2), see blue line segments in Fig. 2c. If such vertices are directly visible from the point p_1 or p_2, an alternative shorter path may be constructed.



(a) (b) (c)

Fig. 2: Approximate paths between two points: (**a**) a rough path; (**b**) a refined path; (**c**) an alternative refined path.

Determination of the cell is the point–location problem, which can be solved in $O(\log N_W)$ or in the average complexity $O(1)$ by the "bucketing" technique [12]. A cell of the node after its adaptation towards the city can be determined during the node movement along a path to the city by a procedure similar to the straight walk in a triangulation [11]. Such cell determination can be bounded by $O(\log n_d)$, where n_d is the number of passed diagonals of the convex partition by the path.

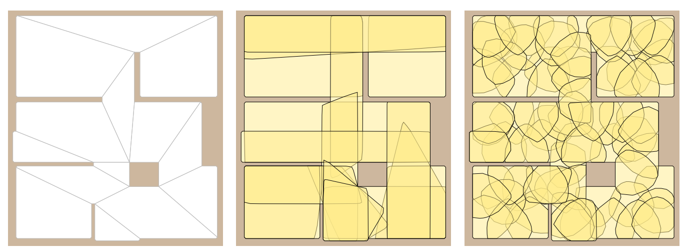
The complexity of the described approximation can be worse than the complexity of the SPM construction, because direct visibility test for each vertex is performed independently on the previous tests. The complexity of the primal path determination can be bounded by $O(n_c^2)$, where n_c is the maximal number of vertices of a convex cell. The refinement procedure depends on the number of path vertices and can be bounded by $O(n_d l)$, where n_d is the number of diagonals involved in the convex partition and l is the maximal number of vertices of a path between two vertices. In practice, the complexity is not so pessimistic, because nodes are moved toward the city, which means winner nodes and particular cities become directly visible during the adaptation. Also a path between two nodes is computed for two neighbouring nodes that are typically close to each other, which means they are both in the same cell or in the next cell.

IV. WATCHMAN ROUTE PROBLEM - WRP

The main idea of the proposed adaptation procedure for the WRP is based on the TSP procedure presented in the previous section, where a ring of nodes evolves in the polygonal domain $\mathcal{W}$. The ring evolution can be viewed as an "exploration" of the polygonal domain $\mathcal{W}$ and the ring may represent a watchman route. Covered parts of $\mathcal{W}$ from the route (denoted as the ring coverage) can be computed during the adaptation and instead of cities presented to the network in the TSP, representative points of uncovered parts of $\mathcal{W}$ can be used to attract nodes towards the uncovered parts in order to find a route from which whole $\mathcal{W}$ will be covered. An important aspect of the uncovered parts should be considered: to cover a convex part of $\mathcal{W}$ it is sufficient if the route just enters into the part. From this perspective, particular representative point of the part is more like an attraction point towards which a node is moved, thus these points are called attraction points. To support the idea of the proposed adaptation it is necessary to address three sub-problems:

1) determination of the current ring coverage and uncovered parts of $\mathcal{W}$,
2) determination of attraction points of the uncovered parts,
3) adaptation of nodes to attraction points.

Proposed solutions of these sub-problems are based on a convex cover set of $\mathcal{W}$ and a triangular mesh of $\mathcal{W}$. The WRP is considered as a problem to find a route that is incident with a subset of convex polygons from the cover set such that the union of these polygons covers $\mathcal{W}$. Convex polygons of the cover set are restricted to respect the d-visibility, i.e. a distance of two points of a convex polygon is less than d. Examples of a convex partition and convex cover sets are shown in Fig. 3.



(a) partition (b) cover set (c) cover set, d=1 m

Fig. 3: A convex polygon partition and cover sets; (**c**) cover set according to restricted visibility range d. Polygons of cover sets are visualized in a semitransparent light (yellow), thus overlapping parts of the polygons are darker (stronger yellow).

A triangular mesh is used to find a cover set, despite any cover set can be possibly used. The mesh is advantageous, because it can be used to support determination of the ring coverage. Moreover, a centroid of each triangle can be used as an attraction point. The mesh generator [31] provides a triangular mesh with the specified maximal triangle area and minimal triangle angle. Appropriate parameters of the mesh have to be selected according to visibility range d. More triangles provide better results, but they also increase the computational burden. An important aspect of the mesh is its quality, triangles should be equilateral (close to be equilateral) or the Delaunay property must be satisfied, otherwise issues with degenerative cases can be expected [11].

A convex cover $\boldsymbol{P}$ is a set of convex polygons $\boldsymbol{P} = \{P_1, \ldots, P_{N_C}\}$, where each convex polygon P_i contains a subset of triangles of the supporting triangular mesh $\mathcal{T} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{T})$. These triangles are used to determine the current ring coverage. An algorithm to find a cover set is described in the next section followed by a procedure of the ring coverage determination, Section IV-B. The adaptation procedure for the WRP with d-visibility is presented in Section IV-C.

A. Finding a Cover Set

A convex polygon P of a cover set is found as a convex hull of mesh triangles' vertices. Each polygon P is formed from the mesh vertices $\boldsymbol{V}_P$, $\boldsymbol{V}_P \subseteq \boldsymbol{V}$ and has associated set of mesh triangles $\boldsymbol{T}_P$, $\boldsymbol{T}_P \subseteq \boldsymbol{T}$ that are entirely inside P, $T \in \boldsymbol{T}_P, T \subseteq P$. The procedure to find a convex polygon P is depicted in Algorithm 1, where $\boldsymbol{V}(T)$ represents the mesh vertices of the triangle T and $\boldsymbol{E}(T)$ are edges of T. A construction of the convex hull is initiated from a (possibly

Algorithm 1: Find Convex Polygon

Input: $\mathcal{T} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{T})$ – a triangular mesh of $\mathcal{W}$
Input: d – a visibility range
Input: T_r – an initial triangle, $T_r \in \boldsymbol{T}$
Output: $P(\boldsymbol{V}_P, \boldsymbol{T}_P)$ – a convex polygon

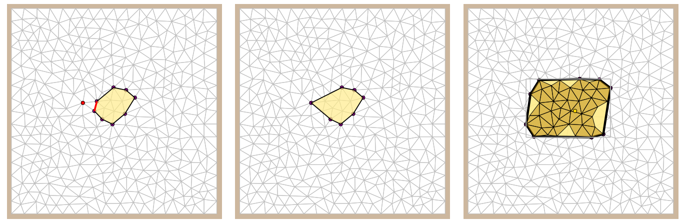
```
V_P ← V(T_r), T_P ← {T_r}
E_open ← {e|e ∈ T_r}, E_close ← ∅
while |E_open| > 0 do
    e* ← random(E_open)                    // random edge
    T* ← T incident with e* ∧ e* ∈ E(T) ∧ T ∉ T_P
    v* ← v such that v ∉ V_P ∧ v ∈ T*
    C_h ← convex_hull(V_P, v*, d)
    if T* is entirely inside C_h then
        V_P ← C_h, T_P ← T_P ∪ {T*}
        E_open ← E_open ∪ {e|e ∈ E(T*) ∧ e ∉ E_close}
    else
        E_close ← E_close ∪ {e|e ∈ E(T*)}
    E_open ← E_open \ {e*}, E_close ← E_close ∪ {e*}
```

random) triangle T_r, which forms an initial convex hull. The hull is extended by mesh vertices that are opposite to edges of the triangle (the set $\boldsymbol{E}_{open}$). Each particular mesh vertex can extend the hull by one triangle. The triangle is associated to the polygon and the convex hull is modified according to the newly added mesh vertex. The procedure is repeated until the list of candidate edges $\boldsymbol{E}_{open}$ is empty. During the hull extension (convex_hull), dimensions of the hull are considered and a mesh vertex being added is eventually discarded. An example of the algorithm performance is shown in Fig. 4.

The algorithm to find a convex polygon is used to find a complete cover set. The set is found by a randomized incremental procedure that selects a random uncovered triangle of the used triangular mesh and extends it to a convex polygon. The procedure is depicted in Algorithm 2.

B. Determination of the Ring Coverage

The ring coverage is computed as approximation of the continuous sensing along the ring that is based on the computation of coverage of $\mathcal{W}$ along a straight line segment s of two directly visible points. The coverage is found as the union of



(a) (b) (c)

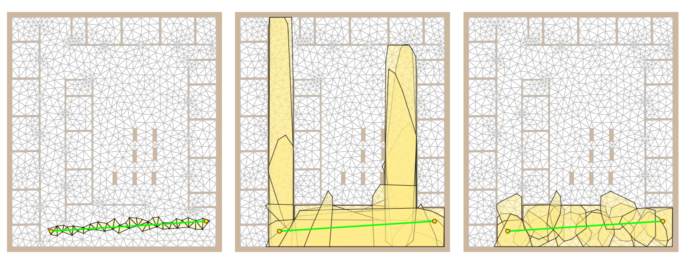
Fig. 4: An example of convex polygon determination: (**a**) a mesh vertex being added to the hull, (**b**) an extented polygon, (**c**) the final convex polygon and associated triangles that are entirely inside the polygon.

Algorithm 2: Find Convex Cover Set

Input: $\mathcal{T} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{T})$ – a triangular mesh of $\mathcal{W}$
Input: d – a visibility range
Output: $\boldsymbol{P} = \{P_1, \ldots, P_{N_C}\}$ – a set of convex polygons

```
U ← T                                      // uncovered triangles
while |U| > 0 do
    T ← select random triangle from U
    P(V_P, T_P) ← find_convex_polygon(𝒯, d, T)
    P ← P ∪ P, U ← U \ T_P
```

all triangles associated to all incident convex polygons of the cover set with the ring. Each triangle is in at least one convex polygon, therefore for each triangle a set of convex polygons (which have associated the triangle) is associated. All incident convex polygons with s are determined from incident triangles, which are found by a visibility walk in the triangular mesh. An example of the segment coverage is shown in Fig. 5. The



(a) incident triangles (b) d = 50 m (c) d = 4 m

Fig. 5: An example of approximation of the continuous sensing along a line segment, triangular mesh with 2 266 triangles: (**b**) and (**c**) incident convex polygons for a visibility range d.

ring coverage is determined in three steps.

1) A sequence of points $(p_1, p_2, \ldots, p_n, p_{n+1})$, where $p_1 = p_{n+1}$ for the closed ring, representing the ring is found by the approximation of the shortest path between each neighbouring nodes of the ring.
2) For each segment of neighbouring points $s_i = (p_i, p_{i+1})$ incident triangles are determined by the visibility walk in the triangular mesh. A set $\boldsymbol{T_r}$ is the union of all such incident triangles.
3) The ring coverage is a set of triangles $\boldsymbol{T_c}$ determined

from the convex polygons. For each $T_i \in \boldsymbol{T_r}$ all associated convex polygons $\boldsymbol{P_i} = \{P_{i,1}, \ldots P_{i,n}\}$ are used to find the set $\boldsymbol{T_c}$, $\boldsymbol{T_c} = \bigcup_{T_i \in \boldsymbol{T_r}} \bigcup_{P_{i,j} \in \boldsymbol{P_i}} \{T | T \in P_{i,j}\}$.

In the worst case all triangles can be intersected by a ring and each convex polygon can have associated all triangles, therefore the complexity of the ring coverage procedure can be bounded by $O(N_T^2)$.

C. Adaptation Procedure for the WRP with d-visibility

The adaptation procedure is based on a triangular mesh and a cover set constructed on top of the mesh. The centroids of the mesh triangles are used as attraction points in similar way like cities in the TSP algorithm. The problem is to find a route to "see" all triangles, therefore it is not necessary to visit all triangles. The adaptation rule is modified to do not place nodes unnecessary close to the attraction point p_a. An alternate point is determined and used instead of p_a, if the node would be closer to p_a than the visibility distance d after the adaptation. The proposed adaptation procedure is depicted in Algorithm 3.

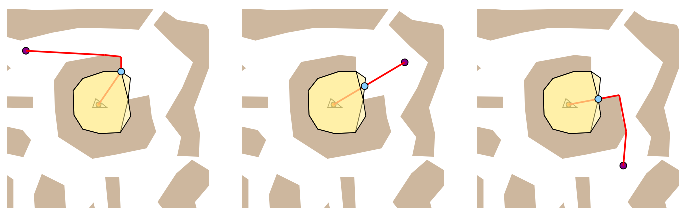
Algorithm 3: WRP Adaptation Procedure

Input: $\mathcal{T} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{T})$ – a triangular mesh of $\mathcal{W}$
Input: $\boldsymbol{P}$ – a set of convex polygons associated to $\boldsymbol{T}$
Input: (G, μ, α) - parameters of SOM
Output: $(\nu_1, \ldots, \nu_m)$ - nodes representing a route

```
r ← initialization            // create a ring of nodes
repeat
    I ← ∅                     // a set of inhibited nodes
    T_c ← triangles covered by the current ring r
    Π(T) ← create a random permutation of triangles
    foreach T ∈ Π(T) do
        if T ∉ T_c then
            p_a ← centroid(T)    // attraction point
            ν* ← select winner node to p_a, ν* ∉ I
            P_c ← {all associated convex polygons to T}
            if ν* ∉ P, P ∈ P_c then
                adapt(ν*, p_a)
            T_c ← T_c ∪ {T|T ∈ P, P ∈ P_c}
            I ← I ∪ {ν*}        // inhibit winner node
        adaptation to triangles
    G ← (1 − α) · G            // decrease the gain
until all triangles are covered by the current ring
```

The algorithm is pretty much similar to the TSP algorithm described in Section III-B. The main difference is in consideration of the coverage and the number of neurons, which is derived from the number of triangles. All other SOM parameters (like G, f, μ, α) are the same. The current ring coverage is represented by a set of triangles $\boldsymbol{T}_c$. At the beginning of each adaptation step the ring coverage is determined. After that, triangles are presented to the network in a random order and nodes are adapted only to uncovered triangles. A winner node is selected according to its distance to the centroid p_a of the presented triangle T. The distance is found as a length of the approximate shortest path from the node to p_a. After the selection, triangles of all associated convex polygons $\boldsymbol{P}_c$ to T are added to $\boldsymbol{T}_c$, which means that the ring is not adapted to these triangles for the rest of the current adaptation step. Also the adaptation is performed only if the winner node is not in some polygon of $\boldsymbol{P}_c$. A node can be in such a polygon due to its movement during adaptation to another triangle.

To avoid placement of nodes unnecessary close to the attraction point p_a, the adaptation rule adapt is modified to find an alternate point for the adaptation. Assume a winner node ν is being adapted to the centroid p_a of the triangle T and T is associated with a set of convex polygons $\boldsymbol{P}_c$. Approximation of the shortest path from ν to p_a can be formed from the map vertices and can be represented as a sequence of points $(p_1, \ldots, p_k)$, where $p_1 = \nu$ and $p_k = p_a$. The alternate point is found as the farthest (from p_a) intersection point of the segment (p_{k-1}, p_k) with $P \in \boldsymbol{P}_c$, see Fig. 6.



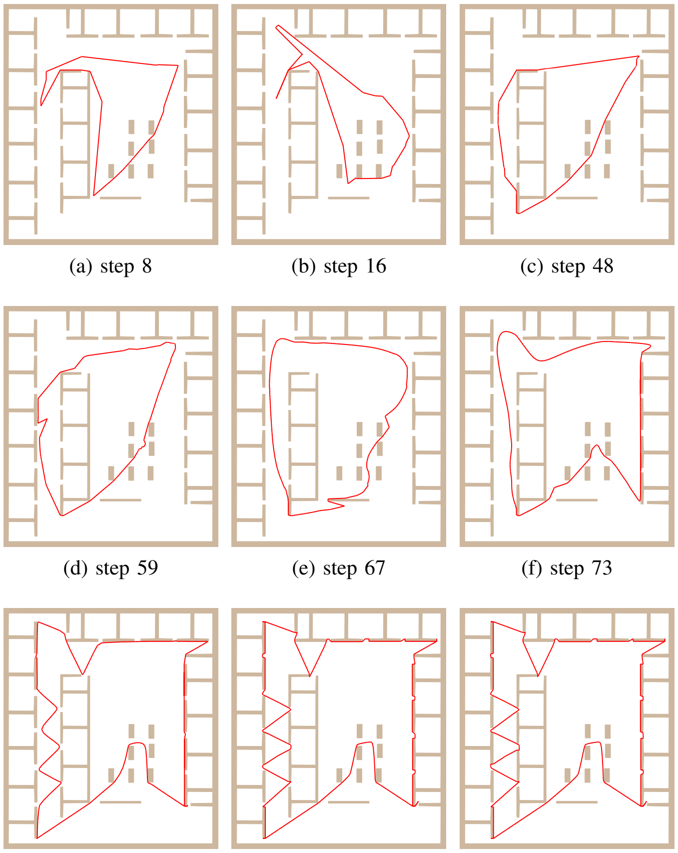
(a) (b) (c)

Fig. 6: An example of alternate points for a different node and the same attraction point, the attraction point is the centroid of the small triangle inside associated convex polygons, line segments represent approximation of the shortest path.

An example of the algorithm performance is shown in Fig. 7. In last thirty steps, a shape of the ring is almost same and winner nodes are moved towards uncovered parts of $\mathcal{W}$, while the ring coverage is preserved.

The complexity of the algorithm depends on the used supporting geometrical procedures that depend on sizes of the map (N_W), triangular mesh (N_T) and convex cover set (N_C). The procedures are called during the adaptation, therefore the complexity of one adaptation step is always related to the number of neurons m. The complexity of the winner selection phase can be bounded by $O(mN_TT_d)$, where T_d is the complexity of the node–point distance query. The adaptation can be bounded by $O(lN_T(T_a + T_m))$, where l is the number of adapted neighbouring nodes (derived from m), T_a is the complexity of the determination of an alternate point, and T_m is the complexity of a node movement towards the attraction (alternate) point along a path. The path is found during the winner selection. The number of neurons is derived from the number of triangles N_T. Thus, for the $N_T \gg N_C$ the complexity of the adaptation step can be bounded by $O(N_T^2 N_W^2)$.

An important aspect of the proposed adaptation procedure relating to the used triangular mesh should be noted. The number of triangles can be relatively high, like for the problem in Fig. 7, where $\mathcal{W}$ represents a map of real building with dimensions about twenty times twenty meters. The high number of



(a) step 8 (b) step 16 (c) step 48

(d) step 59 (e) step 67 (f) step 73

(g) step 81 (h) step 95 (i) step 111

Fig. 7: An evolution of ring during solution of the WRP, map *jh*, triangular mesh with 1 417 triangles, 100 convex polygons for the unrestricted visibility range.

triangles requires more computational time to select a winner nodes, especially for the first adaptation steps. After several steps, the ring covers large portion of $\mathcal{W}$, thus a winner node is determined only for several uncovered triangles. It means that the winner selection is less computationally demanding. Moreover, less number of neurons than in the TSP (according to the number of cities/triangles) is sufficient, because the network is effectively adapted to less number of attraction points. In the experimental evaluation of the algorithm, the number of neurons has been set in a range from $0.1N_T$ to $0.5N_T$.

Despite the fact that the used SOM schema is relatively simple, the proposed adaptation procedure becomes quite complex, because of structures and algorithms to support path and visibility queries. The supporting structures and algorithms are summarized in the following list:

- a convex partition,
- a convex cover set,
- a triangular mesh,
- the visibility graph,
- all shortest paths between vertices of $\mathcal{W}$,
- a point location,
- approximation of the shortest path,
- a straight walking procedure in a convex partition.

On the other side, the advantage of these structures and algorithms is their relative simplicity and computational feasibility.

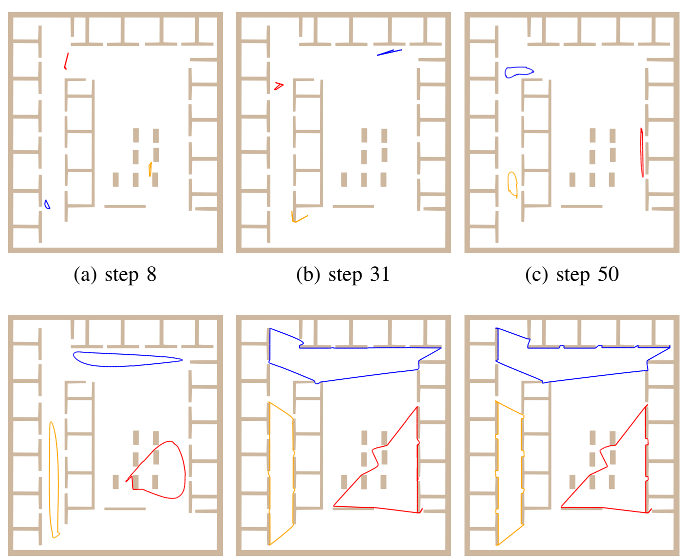
The proposed algorithm addresses the WRP variant that can be found as *d-sweeper route problem* in the literature. An application of the algorithm to address the so-called *d-watchman route problem* is straightforward, only triangles of convex polygons that are connected with the border of $\mathcal{W}$ have to be considered.

V. Multiple Watchmen Route Problem - MWRP

An extension of the proposed WRP algorithm to the MWRP is analogously straightforward like the extension of the TSP algorithm [33] to the MTSP [34], which is described in Section III-B. The main difference is that the original MWRP formulation does not consider a common depot, a solution of the MWRP consists of a set of independent patrolling routes. The flexibility of the SOM approach allows extension to address both MWRP variants with and without the common depot. The MinMax criterion is considered like in the MTSP, i.e. by weighting of the node–attraction point distance to prefer selection of a node from shorter rings. Even though, the winner node is selected according to the attraction point it is then adapted to the alternate point exactly like in the WRP algorithm.

A. MWRP without Common Depot

The algorithm for the variant without the common depot is almost identical to Algorithm 3, the only difference is maintenance of k rings for k watchmen and determination of ring lengths to prefer selection of nodes from shorter rings. An



(a) step 8 (b) step 31 (c) step 50

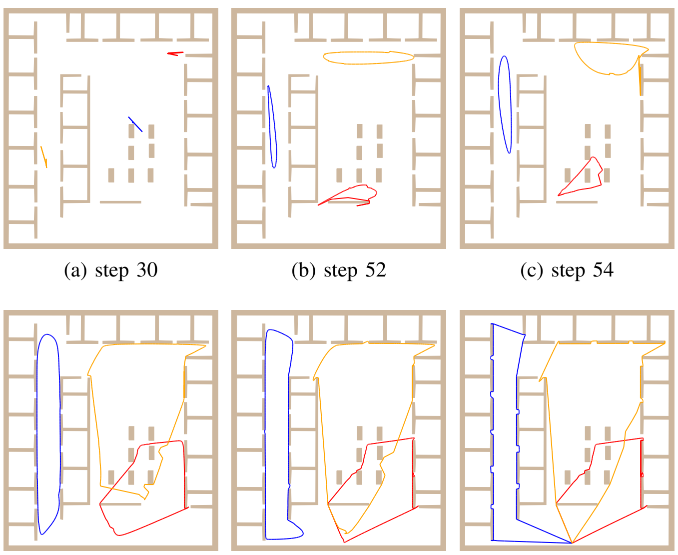
(d) step 54 (e) step 67 (f) step 96

Fig. 8: An evolution of rings in the MWRP without a common depot, map *jh*, triangular mesh with 1 417 triangles, 100 convex polygons for unrestricted visibility range, lengths of found routes are 36.9, 43.8 and 35.3 meters.

example of the algorithm performance is depicted in Fig. 8. It is shown that in the first steps, rings are separated and after additional 50 adaptation steps, the rings are expanded. In the step 67, the solution is almost found, but additional 29 steps are necessary to achieve the complete coverage of $\mathcal{W}$.

B. MWRP with Common Depot

To connect rings with the common depot a winner node from each ring is selected and moved towards the depot at the beginning of each adaptation step before adaptation to uncovered triangles. The alternate point is not determined in adaptation to the depot, because depot is a point that has to be visited. Therefore, the adaptation procedure is terminated if the complete coverage is achieved and the depot winners are closer to the depot than given maximal distance. An example of the algorithm performance is shown in Fig. 9, found routes are of course due to the depot longer than for the MWRP without the depot.



(a) step 30 (b) step 52 (c) step 54

(d) step 62 (e) step 65 (f) step 101

Fig. 9: An evolution of rings in the MWRP with the common depot, map *jh*, triangular mesh with 1 417 triangles, 100 convex polygons for unrestricted visibility range, lengths of found routes are 43, 62 and 58 meters.

VI. Experimental Results

The proposed WRP adaptation procedure has been experimentally verified in a set of environments represented as polygonal maps[1] and selected visibility ranges. Basic properties of the maps are depicted in Table II. The first four environments are maps of real buildings. A supporting triangular mesh is created by the quality mesh generator `triangle` [31] for the minimal triangle angle 32.5° and 25.0° for the map *jh*, and given maximum triangle area. The area is experimentally set according to the circumscribed circle of the triangle, whose radius is derived from the restricted visibility range d. Beside the mesh, a convex polygon partition [29] supports

[1]Maps with all supporting structures are available at http://purl.org/faigl/wrp.

TABLE II: Properties of environments

Map	Dimensions [m × m]	No. vertices	No. holes	No. convex polygons
jh	20.6 × 23.2	196	9	77
pb	133.3 × 104.8	137	3	50
ta	39.7 × 46.8	101	2	46
h2	84.9 × 49.7	1 061	34	476
dense	21.0 × 21.5	288	32	150
potholes	20.0 × 20.0	153	23	75
warehouse	40.0 × 40.0	142	24	83

determination of approximate shortest paths, the number of convex polygons is shown in the last column of Table II.

The *jh*, *ta* and *pb* maps are used in comparison of the proposed WRP algorithm with the reference algorithm. Reference solutions are found as solutions of the decoupled approach (AGP+TSP), see Section III-A. The length of the reference watchman route L_{ref} is found by the Concorde solver [3], which provides exact solution of the TSP. To compare performance of the SOM based algorithm for the TSP and WRP, the AGP+TSP is also solved by the SOM algorithm for the TSP described in Section III-B using the approximate shortest path in $\mathcal{W}$. Both SOM algorithms are randomized, therefore each problem is solved twenty times by each algorithm and found solutions are compared by the percent deviation to the length of the reference route of the mean solution value, $\%PDM = (\overline{L} - L_{ref})/L_{ref} \cdot 100\%$, and as the percent deviation from the reference of the best solution value ($\%PDB$).

Examined algorithms have been implemented in C++ and compiled by the G++ 4.2 compiler with -O2 optimization flags. All experiments have been performed within the same computation environment with the Athlon X2 5050e@2.6 GHz CPU, 2 GB RAM running FreeBSD 7.1, and only one CPU core has been utilized by the algorithms. The presented required computational times do not include creation of all supporting structures. Creation of the convex cover set from the triangular mesh takes a fraction of second for high visibility ranges and less than two seconds for a triangular mesh with seven thousands triangles. The triangular mesh and the convex polygon partition are found in less than one hundred milliseconds, and also supporting visibility graphs are found in a fraction of second. According to the computational time to solve the WRP or the TSP, constructions of these structures are negligible. The most time consuming preparation step is a computation of all shortest path between vertices, the required time is included in the presented T. Here, it should be noted that the TSP algorithm uses node–city path approximation that is more efficient than the node–node query, because all shortest path from the map vertices to the cities are precomputed. Thus, only vertices of a single cell are examined in the node–city path determination for the TSP.

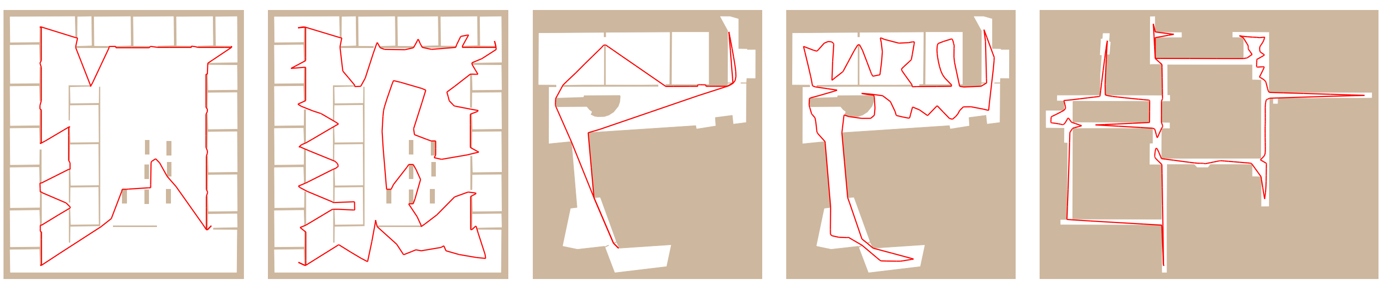
A. Solution of the WRP with d-visibility

Detail experimental results for the WRP with d-visibility are presented in Table III, where N_C, N_T and N_V are properties of the used triangular meshes, m is the number of used neurons

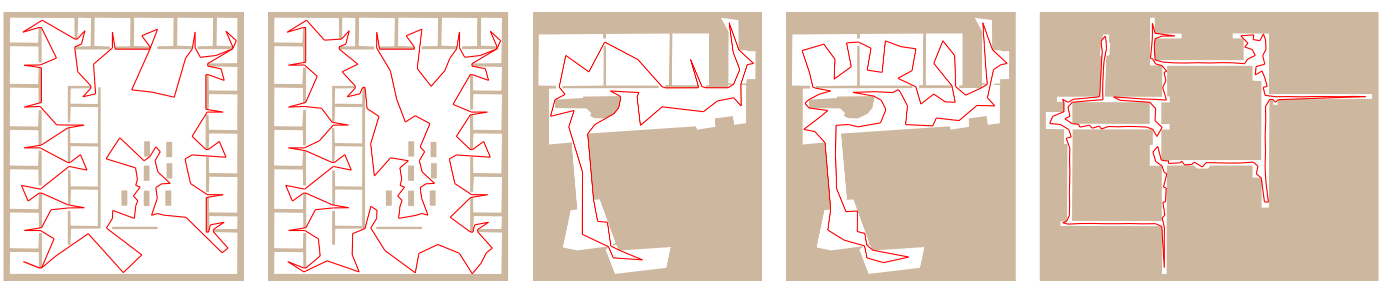
TABLE III: WRP - Results

Map	d [m]	L_{ref} [m]	N_C	N_T	N_V	**SOM - WRP**				**SOM - AGP+TSP**			
						m	**%PDM**	**%PDB**	**T** [s]	m	**%PDM**	**%PDB**	**T** [s]
jh	inf	207.8	100	872	576	88	-50.22	-52.59	6.3	235	1.76	0.59	8.8
jh	10.0	207.3	108	872	576	88	-48.52	-52.82	6.4	238	1.45	0.27	9.1
jh	5.0	216.4	130	872	576	88	-46.94	-50.80	7.1	252	2.34	0.61	10.3
jh	4.0	219.9	169	872	576	175	-32.59	-40.12	18.1	265	1.75	0.19	11.4
jh	3.0	225.5	258	931	608	187	-26.91	-29.46	32.5	288	2.94	1.58	13.4
jh	2.0	281.9	480	1 904	1 183	381	-7.96	-10.58	132.8	438	3.31	1.89	30.3
jh	1.5	350.3	852	2 272	1 392	682	-3.66	-5.24	474.4	705	3.43	1.15	81.0
jh	1.0	470.8	1 800	3 401	1 988	1 701	4.08	2.86	3 277.1	1 380	4.28	3.24	343.8
ta	inf	203.6	46	1 014	638	102	-30.90	-31.60	1.2	85	1.48	0.05	0.9
ta	10.0	202.6	70	1 014	638	102	-28.04	-28.40	1.8	88	1.79	0.05	0.9
ta	5.0	254.1	152	1 014	638	102	-15.36	-17.80	6.4	145	1.09	0.52	2.2
ta	4.0	272.2	209	1 014	638	203	-7.47	-10.00	20.9	180	1.01	0.07	3.3
ta	3.0	315.0	357	1 252	776	376	-5.68	-7.68	73.0	295	1.94	1.41	9.7
ta	2.0	408.3	757	1 944	1 151	778	0.57	-1.13	392.2	578	3.70	3.10	37.6
ta	1.5	522.1	1 320	3 117	1 788	1 247	1.21	0.23	1 232.2	1 012	4.70	3.76	124.4
ta	1.0	743.6	2 955	7 044	3 849	3 522	5.22	3.76	8 587.4	2 162	6.14	5.19	595.5
pb	inf	533.3	52	2 403	1 630	241	-18.58	-20.78	6.2	112	4.00	0.04	1.4
pb	10.0	612.7	111	2 403	1 630	241	-13.03	-14.76	12.1	182	0.56	0.07	3.4
pb	5.0	682.9	262	2 403	1 630	241	-5.81	-9.02	30.3	328	0.42	0.19	11.2
pb	4.0	720.1	373	2 403	1 630	481	-6.75	-8.59	92.3	400	1.52	0.51	17.1
pb	3.0	774.8	714	3 078	2 018	616	-5.81	-6.71	244.0	588	2.34	0.47	36.9
pb	2.0	901.9	1 564	4 692	2 955	1 408	-3.16	-4.05	1 439.8	1 085	2.51	1.27	133.1
pb	1.5	1 115.9	2 787	7 144	4 319	2 858	1.03	0.34	6 181.5	2 052	3.17	2.57	511.7
pb	1.0	1 564.2*	6 188	14 462	8 250	5 785	2.47	1.26	34 090.1	4 522	3.93	3.48	2 544.8

*Due to the high number of cities the solution has been found by the Chained Lin-Kernighan heuristic [4].



(a) WRP, map *jh*, d = *inf*, L = 98.5 m
(b) WRP, map *jh*, d = 3 m, L = 159.1 m
(c) WRP, map *ta*, d = *inf*, L = 139.3 m
(d) WRP, map *ta*, d = 4 m, L = 245.0 m
(e) WRP, map *pb*, d = 4 m, L = 658.2 m

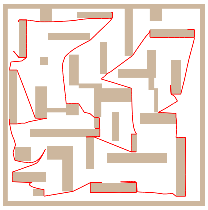


(f) reference, map *jh*, d = inf, L = 207.8 m
(g) reference, map *jh*, d = 3 m, L = 225.5 m
(h) reference, map *ta*, d = inf, L = 203.6 m
(i) reference, map *ta*, d = 4 m, L = 272.2 m
(j) reference, map *pb*, d = 4 m, L = 720.1 m

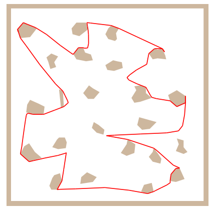
Fig. 10: Selected best solutions found by the proposed WRP algorithm and the reference solutions for a visibility distance d.

and T is the average value of the required computational time in seconds. An overview of the solution quality a histogram of average values is shown in Fig. 12, where 100% represents length of the reference solution. The best solutions of selected problems found by the proposed WRP algorithm and particular reference solutions are shown in Fig. 10. Additional solutions found by the WRP algorithm are depicted in Fig. 11.

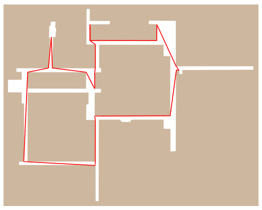
The proposed WRP algorithm provides outstanding solutions for unrestricted and high visibility ranges. The found watchman routes are about tens of percents shorter than the optimal solutions of the related TSP. Even though the solutions are not compared with the exact solutions of the WRP, according to the presented figures one can expect that for high visibility ranges the found solutions are very close to the optima. The WRP algorithm provides overall better results than SOM solutions of the related TSP. Although the



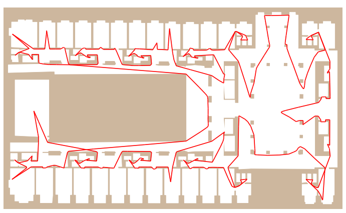
(a) map *warehouse*, d = 10 m



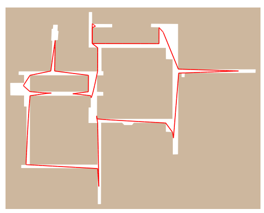
(b) map *potholes*, d = 10 m



(c) map *pb*, d = *inf*



(d) map *h2*, d = 10 m



(e) map *pb*, d = 10 m

Fig. 11: WRP, selected solutions for various visibility range d.

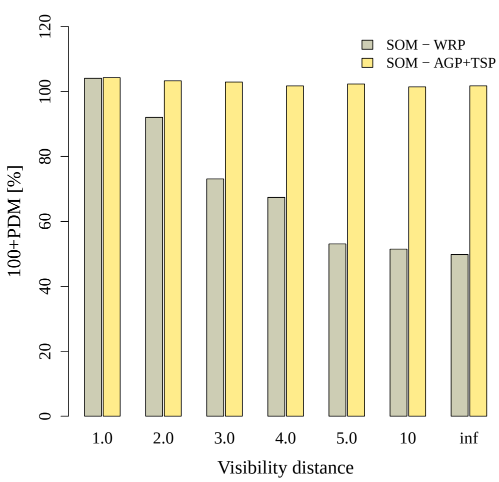


Fig. 12: Average values of the solution quality.

computational burden is higher for the WRP, it has lower memory requires. For the largest problem *pb* and d=1 m, the SOM based TSP algorithm requires hundreds of megabytes (because of the precomputed paths) while the WRP requires about thirty megabytes.

B. Solutions of the MWRP variants

The proposed algorithms for the MWRP have been experimentally verified in a set of problems for various maps and visibility ranges. Examples of found solutions by the proposed algorithm for the MWRP with the common depot are shown in Fig. 13. The performance is not explicitly compared with a reference method, the expected performance of the algorithm should be similar to the SOM based algorithm for the MTSP in the polygonal domain, which is compared with the GENIUS heuristic in [13]. An important aspect of the SOM solutions has to be mentioned. The SOM tries to preserve the topology, thus solutions without crossing tours are more likely found.

If the MWRP is solved without the common depot, independent patrolling routes are found, see Fig. 14.

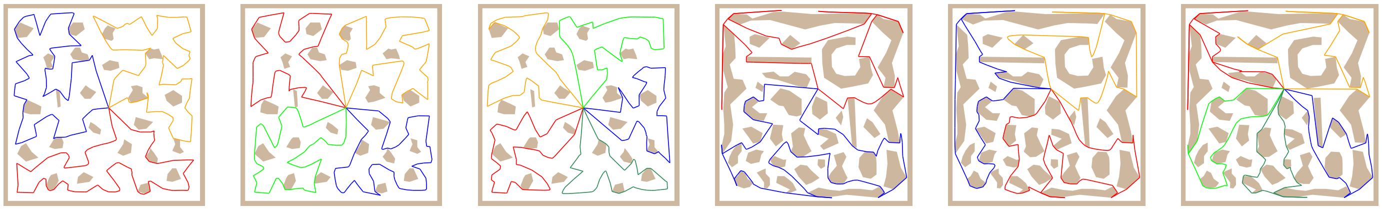
VII. Discussion

The presented experimental results demonstrate feasibility of the proposed SOM algorithm to address the WRP with d-visibility in $\mathcal{W}$. Although the algorithm provides better results than the decoupled approach, the cost of the sensing have to be taken into account. The WRP formulation assumes only the cost of the motion, while the decoupled approach independently minimizes the sensing cost in the AGP part and the motion cost in the TSP part. Authors of [15] noted that the combination of the both costs is a difficult problem that remains largely unexplored. The presented approach can be suitable for such combination, because winner nodes can be considered as guards in the AGP that are selected according to minimization of the route length. Although this idea make sense, it needs further investigation. Nevertheless the outstanding solution quality for high visibility ranges is evident from the results.

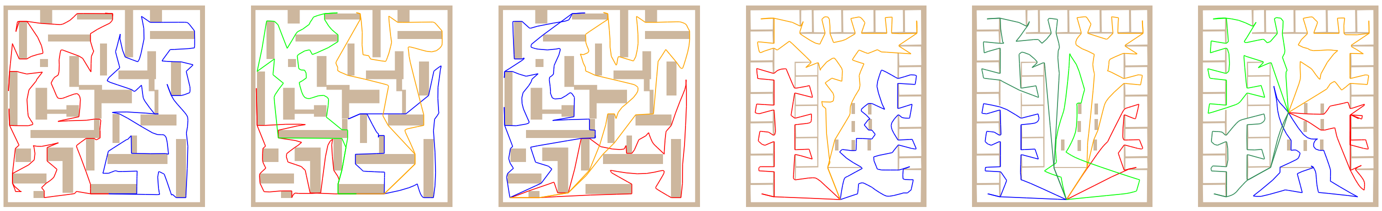
For lower visibility ranges, the solution quality is better than the SOM based solution of the decoupled approach, however it is worse than the exact solution of the TSP part. It can be caused by a poor cover set created from the used triangular mesh or because of the high number of triangles makes the problem very close to the TSP in which SOM provides solutions about units of percents worse than the exact solutions [9]. On the other side, it should be mentioned, that for small d, the *d-sweeper route problem* is very close to the coverage task by a mobile robot, which is addressed by algorithms based on a cell decomposition and explicit routing shapes in the cells [1]. These algorithms provide more suitable solutions for a real robot than approximate (or exact) solutions of the related TSP.

Performance of the MWRP algorithm according to the number of used neurons has been studied during the experimental verification. A following issue has been observed for a relatively small number of neurons. In early adaptation steps, when coverage of the ring is small, the winner nodes have been selected from a longer ring, because all nodes from shorter rings have been inhibited. This issue can be avoided by a creation/deletion of neurons during the adaptation like in one of the first SOM approach for the TSP [2]. However, the deletion has to be carefully considered, because nodes are part of the ring that cover $\mathcal{W}$ and the deletion can change the ring coverage.

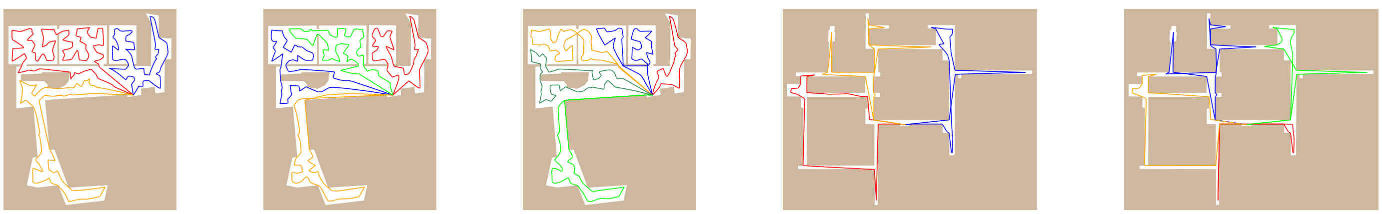
Beside the creation/deletion of nodes, the proposed algorithm can be improved in two aspects. At first, more sophisticated adaptation rules can be used to avoid unnecessary computations of shortest paths, e.g. the local search strategy



(a) map *potholes*, d = 2 m, k = 3 (b) map *potholes*, d = 2 m, k = 4 (c) map *potholes*, d = 2 m, k = 5 (d) map *dense*, d = 4 m, k = 2 (e) map *dense*, d = 4 m, k = 3 (f) map *dense*, d = 4 m, k = 5

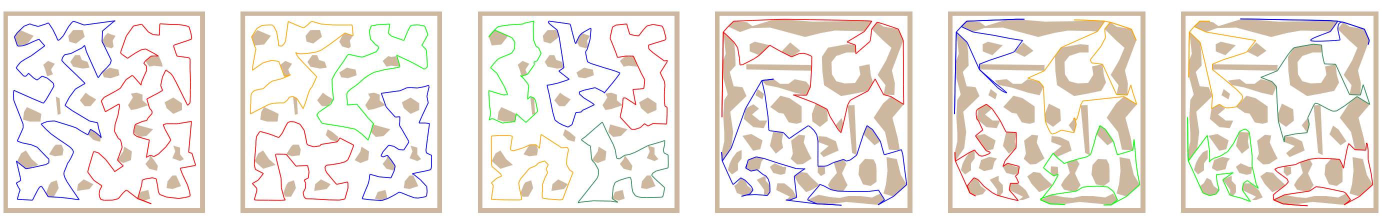


(g) map *warehouse*, d = 4 m, k = 2 (h) map *warehouse*, d = 4 m, k = 4 (i) map *warehouse*, d = 4 m, k = 3 (j) map *jh*, d = 2 m, k = 3 (k) map *jh*, d = 2 m, k = 5 (l) map *jh*, d = 2 m, k = 4

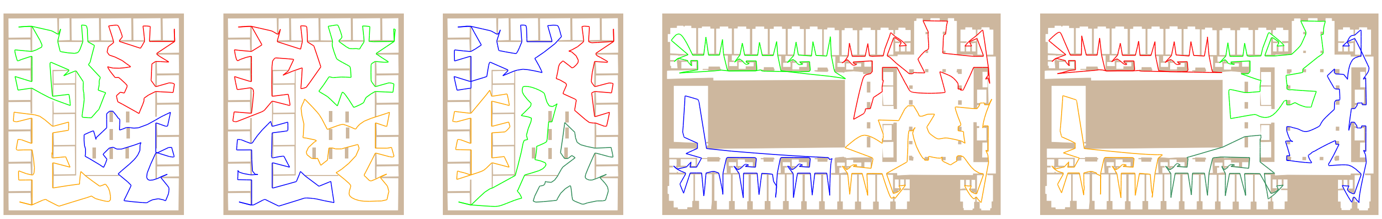


(m) map *ta*, d = 2 m, k = 3 (n) map *ta*, d = 2 m, k = 4 (o) map *ta*, d = 2 m, k = 5 (p) map *pb*, d = 4 m, k = 3 (q) map *pb*, d = 4 m, k = 4

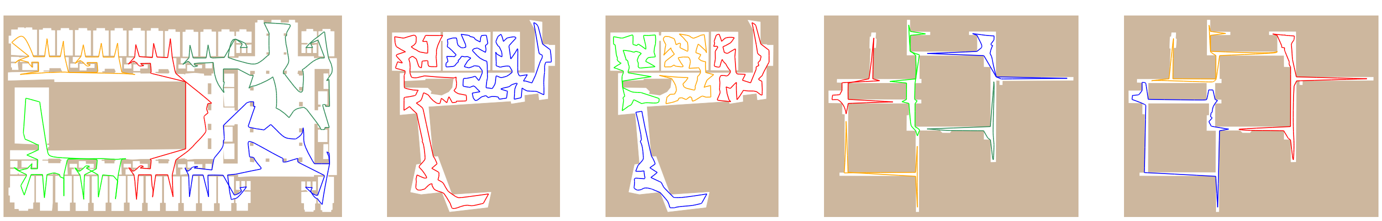
Fig. 13: Selected solutions of the MWRP with the common depot for the d-visibility and k watchmen.



(a) map *potholes*, d = 2 m, k = 2 (b) map *potholes*, d = 2 m, k = 4 (c) map *potholes*, d = 2 m, k = 5 (d) map *dense*, d = 4 m, k = 2 (e) map *dense*, d = 4 m, k = 4 (f) map *dense*, d = 4 m, k = 5



(g) map *jh*, d = 2 m, k = 3 (h) map *jh*, d = 2 m, k = 4 (i) map *jh*, d = 2 m, k = 5 (j) map *h2*, d = 5 m, k = 4 (k) map *h2*, d = 5 m, k = 5



(l) map *h2*, d = 5 m, k = 5 (m) map *ta*, d = 2 m, k = 2 (n) map *ta*, d = 2 m, k = 4 (o) map *pb*, d = 4 m, k = 5 (p) map *pb*, d = 4 m, k = 3

Fig. 14: Selected solutions of the MWRP without the depot for the d-visibility and k watchmen.

to select a winner node proposed in the Co-Adaptive net algorithm [9]. The second aspect is relating to the used supporting geometrical structures, mainly with the properties of the underlying triangular mesh that have to be set appropriately. A mesh created as a topology representing network [21] may be helpful.

VIII. Conclusion

A new SOM based adaptation procedure has been proposed to address the MWRP with d-visibility in the polygonal domain $\mathcal{W}$. The proposed procedure has been experimentally verified in several environments and compared with the decoupled approach. Direct approach to address the MWRP with d-visibility in $\mathcal{W}$ by the SOM adaptation has not been found in the literature, thus the proposed algorithm is probably the first approximate algorithm for this type of problems.

The proposed approach combines the self-organizing principles and supporting geometrical structures to address problems studied in the computational geometry. Four main issues are addressed by the structures: selection/determination of the attraction points, determination of a path in order to select a winner node, a node movement towards the attraction point, and computation of the watchman route coverage.

Regarding to the experimental results it seems that SOM features fit aspects of the so-called hybrid–visibility problems. The proposed algorithm demonstrates applicability of the SOM principles in these problems, thus the principles may be probably applied to other similar problems from the NP class like vision points, touring polygons or safari route problem.

Acknowledgment

The author would like to thank the editor and the reviewers for their valuable suggestions.

References

[1] E. Acar, H. Choset, and J. Y. Lee, "Sensor-based coverage with extended range detectors," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 189–198, Feb. 2006.

[2] B. Angéniol, G. de la C. Vaubois, and J.-Y. L. Texier., "Self-organizing feature maps and the travelling salesman problem," *Neural Networks*, vol. 1, pp. 289–293, 1988.

[3] D. Applegate, R. Bixby, V. Chvátal, and W. Cook, "CONCORDE TSP Solver," http://www.tsp.gatech.edu/concorde.html, July 2010.

[4] D. Applegate, W. Cook, and A. Rohe, "Chained lin-kernighan for large traveling salesman problems," *INFORMS J. on Computing*, vol. 15, no. 1, pp. 82–92, 2003.

[5] N. Aras, I. Altinel, and J. Oommen, "A Kohonen-like decomposition method for the Euclidean traveling salesman problem-KNIES_DECOMPOSE," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 869–890, July 2003.

[6] Y. Bai, W. Zhang, and Z. Jin, "An new self-organizing maps strategy for solving the traveling salesman problem," *Chaos, Solitons & Fractals*, vol. 28, no. 4, pp. 1082 – 1089, 2006.

[7] Y.-J. Chiang and J. S. B. Mitchell, "Two-point euclidean shortest path queries in the plane," in *Proc. 10th Annu. ACM-SIAM Symp. Discrete algorithms*, Baltimore, 1999, pp. 215–224.

[8] W.-P. Chin and S. Ntafos, "Optimum watchman routes," in *Proc. 2nd Annu. ACM Symp. Comput. Geom.*, Yorktown Heights, New York, 1986, pp. 24–33.

[9] E. M. Cochrane and J. E. Beasley, "The co-adaptive neural network approach to the euclidean travelling salesman problem," *Neural Networks*, vol. 16, no. 10, pp. 1499–1525, 2003.

[10] J.-C. Créput and A. Koukam, "A memetic neural network for the euclidean traveling salesman problem," *Neurocomput.*, vol. 72, no. 4-6, pp. 1250–1264, 2009.

[11] O. Devillers, S. Pion, M. Teillaud, and P. Prisme, "Walking in a triangulation," *Internat. J. Found. Comput. Sci.*, vol. 13, pp. 106–114, 2001.

[12] M. Edahiro, I. Kokubo, and T. Asano, "A new point-location algorithm and its practical efficiency: comparison with existing algorithms," *ACM Trans. Graph.*, vol. 3, no. 2, pp. 86–109, 1984.

[13] J. Faigl, "Multi-goal path planning for cooperative sensing," Ph.D. dissertation, Czech Technical University in Prague, 2010.

[14] H. Ghaziri and I. H. Osman, "A neural network algorithm for the traveling salesman problem with backhauls," *Computers and Industrial Engineering*, vol. 44, no. 2, pp. 267–281, 2003.

[15] H. González-Baños, D. Hsu, and J. Latombe, "Motion planning: Recent developments," in *Autonomous Mobile Robots: Sensing, Control, Decision-Making and Applications*, S. Ge and F. Lewis, Eds. CRC Press, 2006, ch. 10.

[16] J. E. Goodman and J. O'Rourke, Eds., *Handbook of Discrete and Computational Geometry*. Boca Raton, FL: CRC Press LLC, 2004.

[17] J. Hershberger and S. Suri, "An optimal algorithm for euclidean shortest paths in the plane," *SIAM J. Comput.*, vol. 28, no. 6, pp. 2215–2256, 1999.

[18] M. Kallmann, "Path Planning in Triangulations," in *Proc. IJCAI Workshop Reasoning, Representation, and Learning in Computer Games*, Edinburgh, Scotland, July 2005, pp. 49–54.

[19] G. Kazazakis and A. Argyros, "Fast positioning of limited visibility guards for the inspection of 2d workspaces," in *Proc. Int. Conf. Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, September 2002.

[20] F. Li and R. Klette, "An approximate algorithm for solving the watchman route problem," in *RobVis*, 2008, pp. 189–206.

[21] T. Martinetz and K. Schulten, "Topology representing networks," *Neural Networks*, vol. 7, no. 3, pp. 507 – 522, 1994.

[22] J. S. B. Mitchell, "Shortest paths among obstacles in the plane," in *Proc. 9th Annu. ACM Symp. Comput. Geom.*, San Diego, 1993, pp. 308–317.

[23] A. Modares, S. Somhom, and T. Enkawa, "A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems," *International Transactions in Operational Research*, vol. 6, pp. 591–606, 1999.

[24] S. C. Ntafos, "Watchman routes under limited visibility," *Comput. Geom.*, vol. 1, pp. 149–170, 1992.

[25] M. H. Overmars and E. Welzl, "New methods for computing visibility graphs," in *Proc. 4th Annu. ACM Symp. Comput. Geom.*, New York, 1988, pp. 164–171.

[26] E. Packer, "Robust geometric computing and optimal visibility coverage," Ph.D. dissertation, Stony Brook University, New York, 2008.

[27] A. Plebe and A. M. Anile, "A neural-network-based approach to the double traveling salesman problem," *Neural Comput.*, vol. 14, no. 2, pp. 437–471, 2002.

[28] W. R. Scott, G. Roth, and J.-F. Rivest, "View planning for automated three-dimensional object reconstruction and inspection," *ACM Comput. Surv.*, vol. 35, no. 1, pp. 64–96, 2003.

[29] R. Seidel, "A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons," *Comput. Geom. Theory Appl.*, vol. 1, no. 1, pp. 51–64, 1991.

[30] T. C. Shermer, "Recent results in art galleries," in *Proc. IEEE*, vol. 80, September 1992, pp. 1384–1399.

[31] J. R. Shewchuk, "Delaunay refinement algorithms for triangular mesh generation," *Computational Geometry: Theory and Applications*, vol. 22, pp. 1–3, 2001.

[32] K. Smith, M. Palaniswami, and M. Krishnamoorthy, "Neural techniques for combinatorial optimization with applications," *IEEE Trans. Neural Netw.*, vol. 9, no. 6, pp. 1301–1318, Nov 1998.

[33] S. Somhom, A. Modares, and T. Enkawa, "A self-organising model for the travelling salesman problem," *Journal of the Operational Research Society*, pp. 919–928, 1997.

[34] ——, "Competition-based neural network for the multiple travelling salesmen problem with minmax objective," *Computers and Operations Research*, vol. 26, no. 4, pp. 395–407, 1999.

[35] X. Tan and T. Hirata, "Finding shortest safari routes in simple polygons," *Information Processing Letters*, vol. 87, no. 4, pp. 179–186, 2003.

[36] P. Wang, "View planning with combined view and travel cost," Ph.D. dissertation, Simon Fraser University, 2007.