



Fast heuristics for large scale covering-location problems[☆]

Luce Brotcorne^a, Gilbert Laporte^{b,*}, Frédéric Semet^{a,c}

^aLAMIH-ROI, Université de Valenciennes, Le Mont Houy, 59304 Valenciennes Cedex 9, France

^bÉcole des Hautes Études Commerciales, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

^cGRIS, Département d'administration de la santé, Université de Montréal, Case postale 6128, succursale "Centre-ville", Montréal, Canada H3C 3J7

Abstract

We propose fast heuristics for large-scale covering-location problems in which the set of demand points is discrete and the set of potential location sites is continuous. These heuristics are compared on a set of 152 real-life instances arising in cytological screening. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Heuristics; Set covering; Location; Graph colouring; Cliques

1. Introduction

Covering problems hold a central place in location theory. In these problems we are given a set of *demand points* and a set of *potential sites* for locating facilities. A demand point is said to be *covered* by a facility if it lies within a prespecified distance of that facility. Using the classification proposed by Daskin [1], we identify two main classes of covering problems: *mandatory coverage problems* which consists of covering all demand points with the minimum number of facilities, and *maximal coverage problems* which consist of covering a maximum number of demand points with a fixed number of facilities; see, e.g., Drezner [2]. Another important distinction relates to the nature of the sets of demand points and potential sites as each can be discrete or continuous, but this distinction is often blurred since continuous sets can sometimes be discretized and some discrete sets with a very large number of elements can be considered as continuous. In most practical applications both sets are discrete, but Schilling et al. [3] report examples in which at least one of the sets is continuous.

[☆]This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

* Corresponding author.

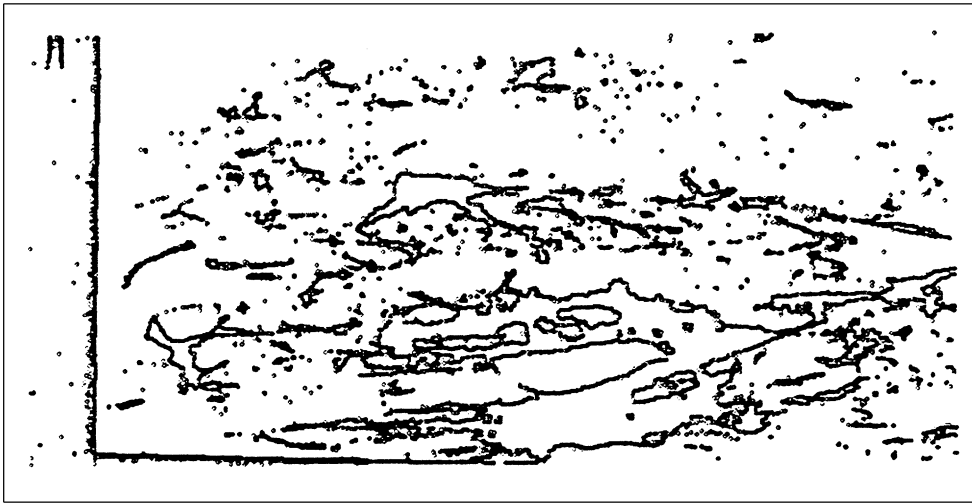


Fig. 1. Cervical specimen.

Our purpose is to describe fast heuristics for mandatory coverage problems in which the set of demand points is very large but discrete, and the set of potential sites is continuous. This work is motivated by a covering problem arising in cytological screening tests (PAP tests) for cervical cancer but has applications in several other areas. In the cytology application, a cervical specimen on a glass slide must be viewed by a screener. An example of such a specimen is depicted in Fig. 1. The screener must be moved to different locations on the glass slide in order to read the entire specimen. To maximize efficiency, the number of viewing locations must be minimized. The area that can be viewed by the screener can be square or circular. For simplicity we will concentrate on the case where the viewing area of the slide must be covered by squares called *tiles*. The square screener moves up and down, or left and right, parallel to the sides of the rectangular glass slide. Locating a tile amounts to determining its center anywhere on the glass slide. The cervical specimen typically appears in the form of smears discretized into pixels. Since there are 625 pixels per square millimeter, the number of demand points can be very large. In our application, it varies between 2500 and 55 000. Another complicating factor of this problem is that it must be solved *in real-time* while the slides are presented to a technician. The only possible approach is to develop heuristics and these must be *very fast*, meaning that a standard problem should be solved rapidly, typically in less than 2 min on a fast Pentium. For this reason, time-consuming local search methods such as simulated annealing or tabu search cannot be applied in this context. The best of the heuristics we have developed can in fact solve these problems in less than 10 s on a Sun Sparcstation 10 (50 MHz) which has approximately half the speed of a 100 MHz Pentium.

Our work has applications in other areas of location such as the siting of wells used to detect contaminants in a ground water system [4], and ambulance location [5]. It is particularly relevant to the latter application in that ambulance deployment decisions must be made quickly, in real-time.

To our knowledge, with one exception, the scientific literature contains no description of any fast heuristic for this type of problem: large-scale mandatory covering problems with a discrete set of

demand points and a continuous set of potential sites. The exception is a previous heuristic developed for the cytological screening problem [6], but our results will show that the best heuristics described on this paper improve vastly upon that presented in [6]. As is done in cytology, our problem will be referred to as the *Tiling Problem*.

The remainder of this paper is organized as follows. In Section 2, we investigate the complexity of the problem and we propose a lower bound. Five new heuristics are described in Section 3. Computational results on a series of real-life cytological problems are presented in Section 4, followed by the conclusions in Section 5.

2. Complexity and lower bound

The Tiling Problem can be shown to be NP-hard. To see this, first consider the decision problem associated with the Tiling Problem: is it possible to cover n demand points in a rectangle using p tiles of side s , parallel to rectangle boundary? To answer this question, consider an instance of a Tiling Problem and construct a tile of side s , parallel to the rectangle boundary, centered at each of the n demand points. The decision problem associated with the Tiling Problem is then equivalent to the following: do there exist p points in the rectangle such that each of the n tiles contains at least one of them? This problem is precisely the *Square Covering Problem* shown to be NP-complete by Megiddo and Supowit [7]. A similar reasoning applies to circle cover, using another result by the same authors.

There exists, however, a polynomially solvable case of the Tiling Problem that occurs when tiles are rectangular and all demand points belong to stripes parallel to the boundary (in either dimension), separated by at least s distance units. In such a case, an optimal solution for each stripe is easily determined by covering the points in a greedy fashion. The proof of optimality is done by induction on the number of points of the stripe. Since no tile overlaps two stripes, this approach is optimal for Tiling Problem. This principle will serve as the basis for one of our heuristics. This heuristic runs in $O(n \log n)$ time if demand points have not previously been sorted, and in $O(n)$ time otherwise.

A lower bound for the Tiling Problem can be obtained from results related to the graph colouring and largest clique problems. For this construct a conflict graph $G = (V, E)$ where the vertex set V corresponds to the set of demand points, and E contains an edge between any two conflicting vertices, i.e., vertices that cannot be covered by the same tile. Then, a lower bound on t^* , the minimum number of tiles necessary to cover the slide, is given by the chromatic number $\chi(G)$ of G , i.e., the least number of colours such that two vertices of G linked by an edge are coloured differently. The size of the largest clique in G is a lower bound on $\chi(G)$ and thus on t^* . There exist graphs for which $\chi(G)$ is strictly inferior to t^* , and graphs for which the maximum clique size is strictly inferior to $\chi(G)$. In the first case consider a circle of unit diameter with three equidistant points A, B and C on its circumference, and a fourth point D outside the circle less than one unit away from A, B and C. Then two circular tiles of unit diameter are necessary to cover all four points, but $\chi(G) = 1$. In the second case, consider a regular pentagon such that any two vertices linked by a diagonal are conflicting, but any two consecutive vertices are not. The conflict graph G corresponding to such a pentagon is depicted in Fig. 2. It can readily be verified that the size of the largest clique in G is 2, but $\chi(G) = 3$.

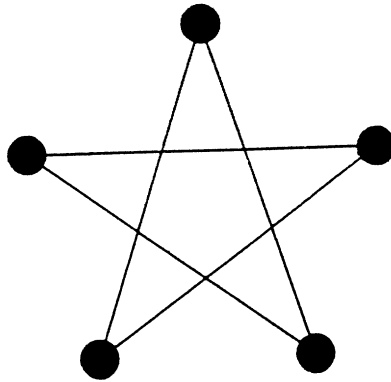


Fig. 2. Conflict graph G for a regular pentagon.

Beyond the fact that the clique lower bound is not tight for the Tiling Problem, solving the maximum clique problem, even heuristically, is impractical for the instance sizes considered in this study. Heuristics exist for large-scale maximum clique problems (see, e.g., [8]), but these do not guarantee tight bounds and have a highly erratic behavior. They certainly cannot be used to assess the quality of a Tiling Problem heuristic.

3. Heuristics

The five heuristics we have developed for the TP apply two or three of the following phases: (1) First solution, (2) set covering, (3) improvement.

3.1. First solution (F1, F2 and F3)

Three variants have been developed for the construction of an initial solution: F1, F2, and F3.

Variant F1: Determine the point P lying at the intersection of the left and bottom boundaries of all demand points. Superimpose on the rectangle a mesh of tiles of side s whose bottom left corner coincides with P . Remove all empty tiles from the mesh.

Variant F2: Sort the demand points by lexicographic order of their abscissae and ordinates. Construct a mesh as in F1. Here this mesh is used not to provide an initial tiling solution, but to define a neighbourhood structure on the points. More specifically, we are interested in six of the tiles associated with any demand point: the tile to which it belongs, called the *home tile*, its top and bottom neighbours, and the three tiles lying immediately to the left of the home tile, and of its top and bottom neighbours (see Fig. 3).

The five tiles surrounding the home tile are called *neighbour tiles*. Then, using the demand points with the lowest abscissa, gradually construct degenerate tiles. At this point these consist of only a vertical side with a height not exceeding s (see Fig. 4). Then considering each other demand point in turn, attempt to integrate it to a tile overlapping its home tile, or to a tile overlapping one of its neighbour tiles, following the order A, B, C, D, E shown in Fig. 3. Integrating a point to an existing

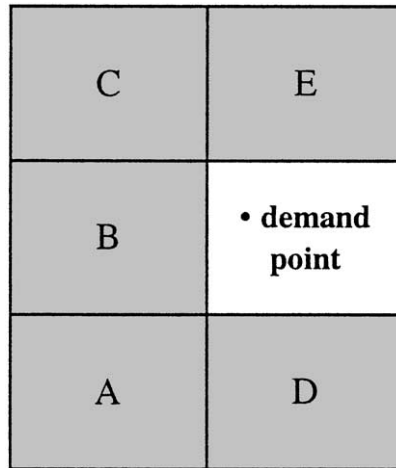


Fig. 3. Home tile (white) and neighbour tiles (shaded).

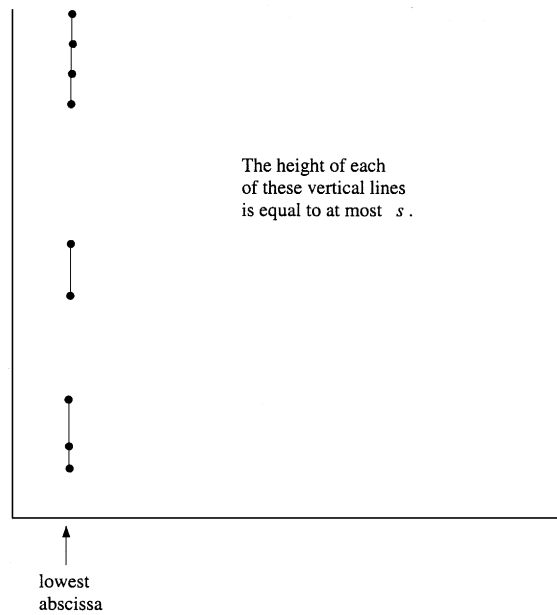


Fig. 4. Initial degenerate tiles.

tile is feasible only if the dimension of the resulting tile does not exceed $s \times s$. Whenever an integration is feasible, the tile is extended to include the new point (see Fig. 5). If the current point cannot be integrated into an existing tile, a new tile is initiated.

Note that throughout this process, tiles are not necessarily square, but they are always rectangles of sides not exceeding s . When all demand points have been considered and are part of a tile, then all tiles are extended so that they become squares of side s .

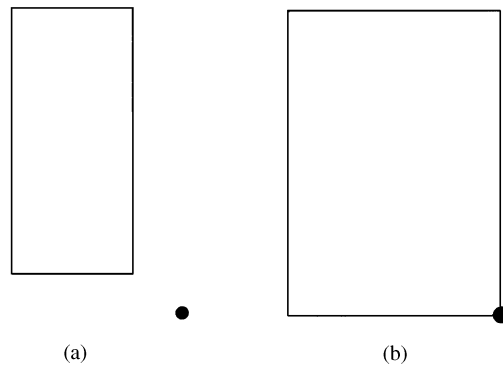


Fig. 5. Feasible extension of an existing tile.

Variant F3: Starting from the left boundary of all remaining demand points, construct a vertical stripe of width s and repeat this procedure until no point remains uncovered by a stripe. Then apply the polynomial algorithm of Section 2 to each vertical stripe separately.

3.2. Set covering (S)

Given an initial solution, determine in each existing tile the smallest rectangle $p_1p_2p_3p_4$ containing all points. Then construct four new tiles including the points and having a vertex at p_1, p_2, p_3 or p_4 . This generates a pool of candidate tiles. A tiling solution can then be obtained from this pool by solving a set covering problem. In our implementation, this is solved using the Balas and Ho [9] PRIMAL1 heuristic. Since no cost is assigned to tiles, corresponding to variables in the set covering problem, this amounts to selecting the tiles in a greedy fashion in non-increasing order of the number of yet uncovered demand points they cover.

3.3. Improvement (I)

The improvement procedure attempts to determine a more economical tiling solution requiring fewer tiles. It is applied four times, from the same solution, using in turn each of the four orientations of the rectangular area. The best overall solution is finally selected. For a given orientation of the rectangle, existing tiles are considered according to the lexicographic order of their bottom left corner. For a tile under consideration, first disregard all its points previously covered by another tile within the improvement procedure. If no point remains within that tile, the tile is then eliminated. Otherwise, construct the smallest rectangle containing all the remaining points of the current tile and extend this rectangle to cover as many points as possible, first in the upward then in the right direction as long as its sides do not exceed s . If either side of the rectangle is strictly less than s , extend it fully so that it becomes a square of side s . This procedure amounts in fact to shifting the tile so that it does not have to overcover some points, and in the hope that it will cover points of some neighbouring tiles. The improvement process is illustrated in Fig. 6 for the case where one tile is eliminated.

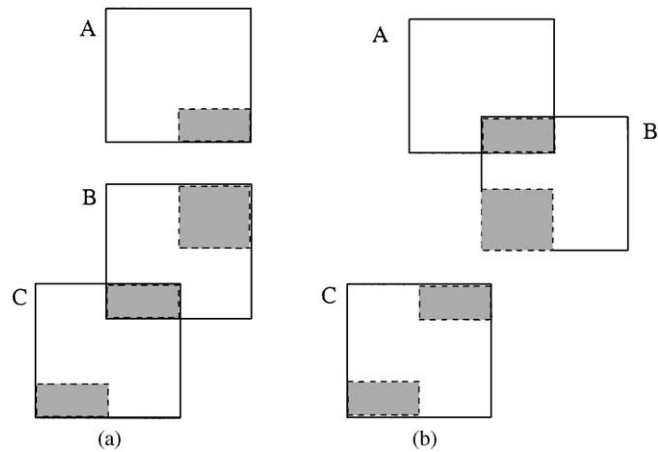


Fig. 6. Improvement procedure. Demand points are represented by shaded areas.

Table 1
Summary of six heuristics

Heuristic	First solution	Set covering	Improvement
H0	F1	S	—
H1	F1	S	I
H2	F2	S	I
H3	F1	—	I
H4	F2	—	I
H5	F3	—	I

3.4. Description of the five new heuristics

Five new heuristics H1–H5 were defined and tested by using different combinations of the above procedures. These heuristics and the procedure H0 used in Laporte et al. (1998) are summarized in Table 1.

To illustrate some of the procedures used in the new heuristics, we depict in Fig. 7 the effect of applying F1, F2 and F1 + I to the data set of Fig. 1.

3.5. Adaptation to circle covers

In some applications using Euclidean distances, demand points are naturally covered by circles instead of squares. Circles are also used in some versions of the cytological screening problem. Then our heuristics can be applied with some modifications. Let r be the radius of the circle. To

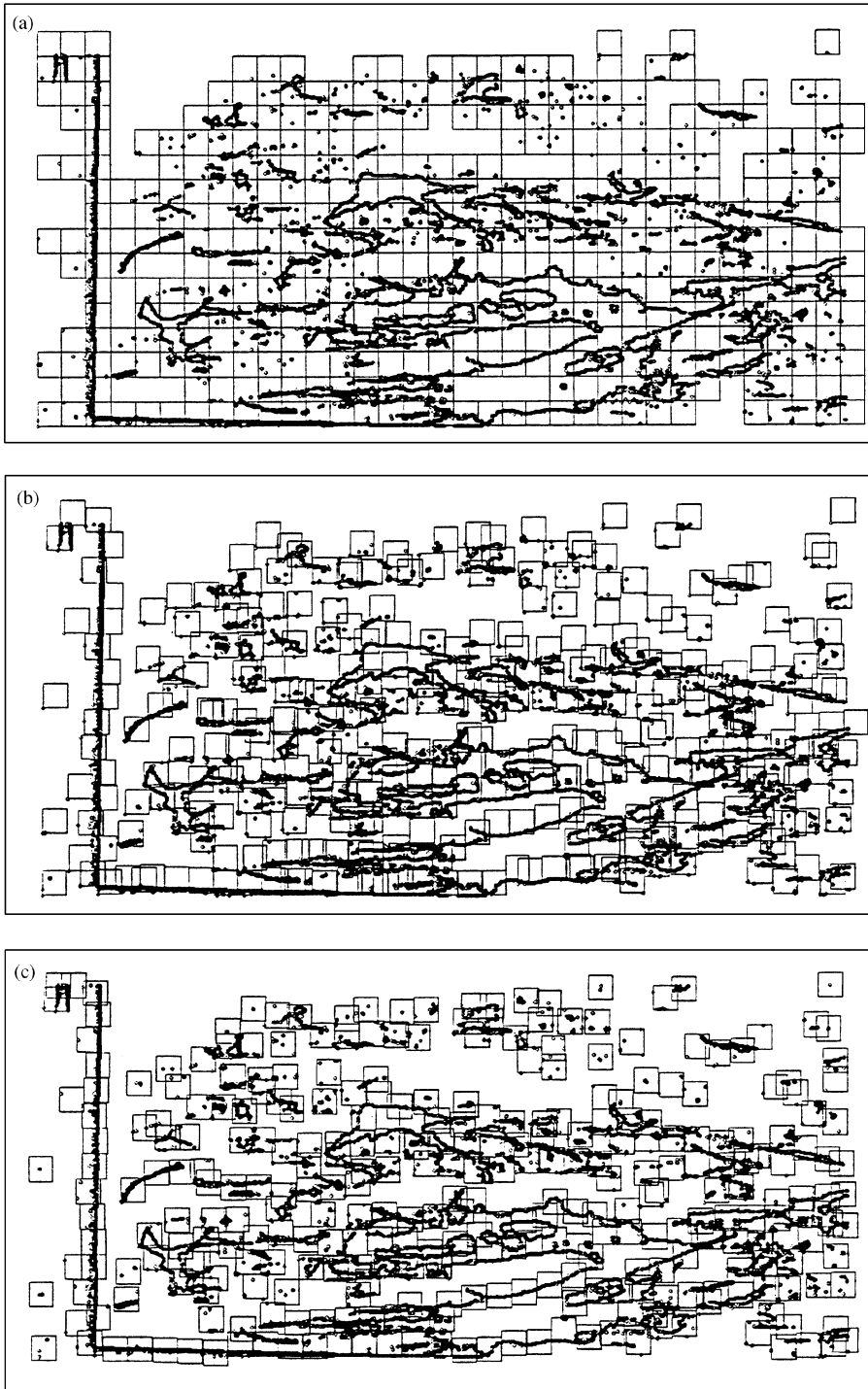


Fig. 7. Illustration of F1, F2 and F1 followed by I on the data set of Fig. 1.

construct the first solution, one can work as in F1, F2 or F3 with inscribed squares of side $s = r\sqrt{2}$ and then replace each square with its circumscribed circle. In the set covering phase, rectangles are constructed using the points of each circle and they are then extended into squares of side $r\sqrt{2}$. The set covering algorithm works with the circles circumscribing these squares. In the improvement phase, we first consider circles in lexicographic order of their centers. All points of a circle already covered are again disregarded. The smallest rectangle covering all remaining points of the circle is considered and possibly extended to cover as many points as possible, only if its sides do not exceed those of the inscribed square. In such a case, the same procedure as in Section 3.3 is applied and all square tiles obtained at the end of this phase are replaced by their circumscribed circle.

- (a) First solution obtained with F1,
- (b) First solution obtained with F2,
- (c) Final solution obtained with F1 followed by I.

4. Computational results

The heuristics just described were coded in C and first run on 152 real-life cytological examples ranging from 2623 to 55 099 demand points. We also created from these examples 152 artificial instances possessing the disjoint stripe property described in Section 2, and thus solvable in polynomial time. These instances were obtained by removing points from the original data sets in order to leave an empty stripe between two consecutive stripes of demand points. They contain between 1159 and 28 659 demand points, approximately half the number of points contained in the original data set. Since these modified examples can be solved to optimality, it becomes easier to assess the value of the heuristics. The original instances were solved using both square tiles and circles; for the modified examples we only used squares since this is the only case for which we can prove that the polynomial algorithm yields an optimal solution. In these instances the size of the glass slide is 1364×640 , the size of the square tile is 41×41 , and the diameter of the circular tile is 56.

The results corresponding to these three problem classes are summarized in Tables 2–4. We report results for heuristics H1–H5 described in Section 3. In addition, for comparison purposes, we provide statistics obtained with the heuristic H0 of Laporte et al. [6]. As mentioned in Section 2, meaningful comparisons with a lower bound cannot be obtained from the current state of knowledge. For each heuristic, we provide the following average statistics over the number of instances of each size class:

Ratio: Ratio of the objective function value (number of square tiles or circles) over the best known value;

Seconds: CPU time in seconds on a Sun Sparc 10 station.

The minimum, average and maximum statistics for all instances are also provided at the bottom of each table.

Results for square tile cover, presented in Table 2, show that in terms of solution quality, H3 has the best performance, followed by H1, then, almost equally, by H2, H4 and H5, and finally by H0. However, a closer examination reveals that H3 is the best heuristic in most cases, except on the smaller instances ($n < 10\,000$) where H2 and H4 have a better performance. In terms of computing time, the ranking is H5, H4, H3, H2, H0 and H1, but there is not much to choose from between H5,

Table 2
Summary of computational results for cytological instances with square tile cover

Instance size	Number of instances	H0		H1		H2		H3		H4		H5	
		Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.
$n < 10\,000$	20	1.098	6.9	1.015	7.4	1.005	1.4	1.014	1.0	1.009	1.0	1.037	1.0
$10\,000 \leq n < 20\,000$	63	1.113	24.8	1.015	25.7	1.013	5.8	1.007	1.3	1.013	1.0	1.027	1.0
$20\,000 \leq n < 30\,000$	41	1.111	45.8	1.013	47.1	1.030	11.8	1.002	2.4	1.030	1.8	1.019	1.6
$30\,000 \leq n < 40\,000$	20	1.097	62.3	1.013	64.6	1.045	17.2	1.001	3.8	1.045	2.9	1.014	2.4
$40\,000 \leq n$	8	1.085	79.1	1.012	82.3	1.071	21.9	1.001	5.5	1.067	4.5	1.004	3.5
Minimum		1.047	1.0	1.000	1.0	1.000	1.0	1.000	1.0	1.000	1.0	1.000	1.0
Average	152	1.107	35.9	1.014	37.1	1.023	9.2	1.005	2.1	1.024	1.6	1.023	1.5
Maximum		1.168	94.0	1.045	97.0	1.097	25.0	1.045	6.0	1.100	5.0	1.071	4.0

Table 3
Summary of computational results for cytological instances with circular tile cover

Instance size	Number of instances	H0		H1		H2		H3		H4		H5	
		Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.
$n < 10\,000$	20	1.106	26.8	1.007	27.6	1.012	8.3	1.024	1.2	1.036	1.6	1.060	1.5
$10\,000 \leq n < 20\,000$	63	1.136	98.9	1.008	101.3	1.013	36.9	1.009	3.2	1.033	5.8	1.057	4.5
$20\,000 \leq n < 30\,000$	41	1.152	161.2	1.012	165.3	1.021	60.9	1.002	5.5	1.040	11.0	1.059	8.0
$30\,000 \leq n < 40\,000$	20	1.179	170.2	1.013	176.4	1.031	75.6	1.000	8.1	1.047	18.3	1.073	11.7
$40\,000 \leq n$	8	1.207	115.3	1.007	124.3	1.049	66.0	1.001	11.1	1.052	26.8	1.088	15.9
Minimum		1.057	6.0	1.000	7.0	1.000	1.0	1.000	1.0	1.000	1.0	1.018	1.0
Average	152	1.146	116.5	1.010	119.9	1.020	46.2	1.008	4.6	1.038	9.4	1.062	6.6
Maximum		1.223	303.0	1.055	309.0	1.059	113.0	1.056	13.0	1.074	31.0	1.100	18.0

Table 4
Summary of computational results for modified cytological examples (stripes) with square tile cover

Instance size	Number of instances	H0		H1		H2		H3		H4		H5	
		Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.
$n < 10\,000$	65	1.087	9.5	1.022	9.7	1.038	1.6	1.043	1.0	1.053	1.0	1.000	1.0
$10\,000 \leq n < 20\,000$	77	1.082	38.4	1.021	38.9	1.047	5.5	1.060	1.0	1.063	1.0	1.000	1.0
$20\,000 \leq n < 30\,000$	10	1.063	102.3	1.010	103.1	1.051	12.3	1.048	1.9	1.062	1.3	1.000	1.0
Minimum		1.028	1.0	1.000	1.0	1.009	1.0	1.000	1.0	1.012	1.0	1.000	1.0
Average	152	1.083	30.8	1.021	31.2	1.044	4.3	1.052	1.1	1.059	1.0	1.000	1.0
Maximum		1.137	139.0	1.055	140.0	1.093	19.0	1.119	2.0	1.116	2.0	1.000	1.0

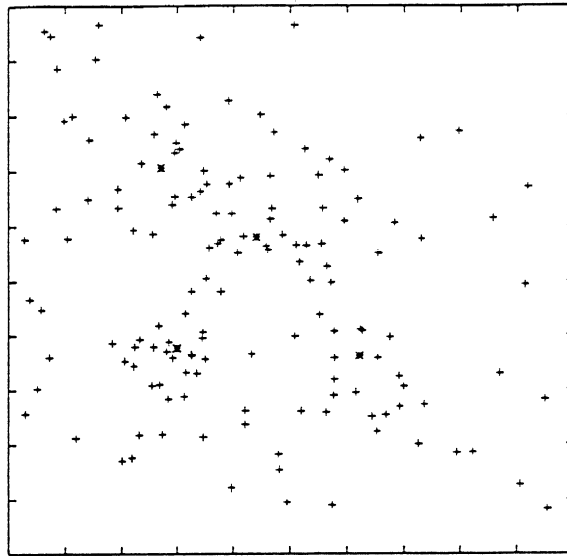


Fig. 8. Generation of instances according to the Cordeau et al. [10] procedure. Seed points *; other points + .

H4 and H3. As a rule, the set covering phase is the most time consuming, as seen by the computing times of H0, H1 and H2. This time is smaller in the case of H2 because this heuristic generates fewer potential tiles for the set covering problem. Our recommendation is therefore to use H3 for $n \geq 10\,000$ and H2 or H4 for $n < 10\,000$.

Computing times increase significantly when circles are used instead of squares since more operations are then required. Results shown in Table 3 indicate that in the case of circles, the relative ranking of the six heuristics is H3, H1, H2, H4, H5 and H0. In terms of computing times, the ranking is H3, H5, H4, H2, H0 and H1. Again our recommendation is to use H3 except possibly when $n < 10\,000$, in which case H2 is probably the best choice if both solution quality and computing time are taken into account.

Results presented in Table 4 indicate that for stripe instances, H5 always provides the optimum in about 1 s, irrespective of n , and H1 typically provides solutions within 2.2% of optimality.

In addition to the tests carried out on the cytological examples, we have also performed some computations on randomly generated instances. A first set was obtained by generating n points with integer coordinates in the 1000×1000 square, according to a discrete uniform distribution. A second set was generated as in Cordeau et al. [10], by first randomly generating k integer coordinate seed points in the 1000×1000 square, and then n integer coordinate points, according to the following procedure governed by an input parameter ϕ :

Step 1: Set $i := 1$.

Step 2: While $i \leq n$, do

- (a) Randomly generate a point v_i in the 1000×1000 square according to a discrete uniform distribution, and compute its distance d to the nearest seed point.

Table 5
Summary of computational results for instances with uniformly generated points and square tile cover

Instance size	Number of instances	H0		H1		H2		H3		H4		H5	
		Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.
$n = 10\,000$	10	1.122	34.4	1.011	35.0	1.070	14.5	1.008	1.0	1.071	1.0	1.000	1.0
$n = 20\,000$	10	1.086	42.8	1.022	44.1	1.084	19.8	1.019	2.0	1.083	2.0	1.000	1.0
$n = 30\,000$	10	1.085	55.7	1.032	57.7	1.095	22.3	1.031	3.0	1.096	3.0	1.000	2.0
Minimum		1.083	26.0	1.002	27.0	1.056	10.0	1.000	1.0	1.058	1.0	1.000	1.0
Average	30	1.098	44.3	1.022	45.6	1.083	18.9	1.019	2.0	1.083	2.0	1.000	1.3
Maximum		1.129	62.0	1.038	64.0	1.104	28.0	1.035	3.0	1.108	3.0	1.004	2.0

Table 6
Summary of computational results for randomly generated instances with a high concentration around seed points and square tile cover

Instance size	ϕ	Number of instances	H0		H1		H2		H3		H4		H5	
			Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.	Ratio	Sec.
10 000	0.04	10	1.132	34.7	1.013	35.3	1.047	12.9	1.001	1.0	1.050	1.0	1.009	1.0
	0.06	10	1.134	36.6	1.013	37.2	1.028	12.3	1.000	1.0	1.033	1.0	1.012	1.0
	0.08	10	1.148	45.1	1.025	45.9	1.022	13.1	1.001	1.0	1.024	1.0	1.009	1.0
20 000	0.04	10	1.097	53.4	1.018	55.1	1.071	19.8	1.011	2.0	1.073	2.0	1.000	1.0
	0.06	10	1.099	53.7	1.018	55.3	1.059	20.6	1.002	2.0	1.059	2.0	1.001	1.0
	0.08	10	1.114	54.1	1.016	55.6	1.054	20.2	1.000	2.0	1.056	2.0	1.004	1.0
30 000	0.04	10	1.084	64.7	1.021	66.9	1.073	20.9	1.016	3.0	1.078	3.1	1.000	2.0
	0.06	10	1.091	63.2	1.022	65.4	1.080	20.7	1.012	3.0	1.082	3.1	1.000	2.0
	0.08	10	1.092	62.8	1.017	65.0	1.065	20.6	1.006	3.0	1.066	3.1	1.000	2.0
Minimum			1.076	31.0	1.000	31.0	1.005	10.0	1.000	1.0	1.012	1.0	1.000	1.0
Average		90	1.110	51.7	1.018	53.1	1.055	18.2	1.006	2.0	1.058	2.0	1.004	1.3
Maximum			1.172	73.0	1.039	75.0	1.091	29.0	1.025	4.0	1.095	4.0	1.024	2.0

- (b) Let u be a number randomly chosen in the $[0, 1]$ interval according to a continuous uniform distribution. If $u < e^{-\phi^d}$, set $i := i + 1$. Otherwise delete v_i .

This generation procedure produces instances in which the point concentration around the seeds is higher (see Fig. 8), as observed in several real-life geographical settings. A smaller value of ϕ produces a smaller concentration.

All randomly generated instances had $n = 10\,000$, $20\,000$ and $30\,000$ points. For the second generation mode $k = n/100$ seed points were used and the values of ϕ were equal to 0.04, 0.06 and 0.08. In all cases, 41×41 square tiles were used. Computational results for these instances are presented in Tables 5 and 6.

Tests results indicate that for the uniform point generation, H5 consistently produces the best results within very short computing times. For instances generated with seed points, H3 is the best heuristic for $n = 10\,000$ and H5 is superior for $n = 20\,000$ or $30\,000$. However, the performance of H3 becomes close to that of H5 when ϕ increases.

5. Conclusions

We have described five new heuristics for large-scale covering-location problems arising in a variety of contexts. In these problems the solution space is continuous in the sense that the set of potential sites is a plane as opposed to a discrete set. Our main example was derived from a problem arising in cytology, but our heuristics apply equally well to other situations such as ambulance location (see, [5]). For cytological instances our best heuristic is H3 for larger cases ($n \geq 10\,000$), and H2 or H4 otherwise. For randomly generated instances, H5 usually produces the best results. As a rule, these heuristics can be executed in only a few seconds, which is what is required for real-time decision making. It is interesting to observe that our best heuristics do away with the set covering algorithm which is widely considered as the standard methodology in covering-location. We believe this conclusion should be kept in mind when solving large-scale covering problems arising in real-life settings.

Acknowledgements

This work was partially supported by the Canadian Natural Sciences and Engineering Research Council grants OGP0039682 and OGP0184123, and by Strategic grant STR0149269. This support is gratefully acknowledged. Thanks are due to the reviewers for their valuable comments.

References

- [1] Daskin MS. Network and discrete location. New York: Wiley, 1995.
- [2] Drezner Z. The p -cover problem. *European Journal of Operational Research* 1986;26:312–3.
- [3] Schilling DA, Jayaraman V, Barkhi R. A review of covering problems in facility location. *Location Science* 1993;1:25–55.
- [4] Meyer PD, Brill Jr. ED. A method for locating wells in a groundwater monitoring network under conditions of uncertainty. *Water Resources Research* 1988;24:1277–82.

- [5] Gendreau M, Laporte G, Semet F. Solving an ambulance location model by tabu search. *Location Science* 1997;5:75–88.
- [6] Laporte G, Semet F, Dadashidze VV, Olsson LJ. A tiling and routing heuristic for the screening of cytological samples. *Journal of the Operational Research Society* 1998;49:1233–8.
- [7] Megiddo N, Supowit KJ. On the complexity of some common geometric location problems. *SIAM Journal on Computing* 1984;13:182–96.
- [8] Homer S, Peinado M. Experiments with polynomial-time CLIQUE approximation algorithms on very large graphs. In: Johnson DS, Trick MA, editors. *Cliques, coloring and satisfiability, DIMACS series in discrete mathematics and theoretical computer science*, Vol. 26. Providence, RI: American Mathematical Society, 1996. p. 147–67.
- [9] Balas E, Ho A. Set covering algorithms using cutting planes, heuristics and subgradient optimization: a computational study. *Mathematical Programming Study* 1980;12:37–60.
- [10] Cordeau J-F, Gendreau M, Laporte G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* 1997;30:105–19.