Local search study of honeycomb clustering problem for cellular planning

Jean-Charles Créput* and
Abderrafiaa Koukam

Systems and Transportation Laboratory,
University of Technology of Belfort-Montbeliard,
90010 Belfort, France
E-mail: Jean-Charles.Creput@utbm.fr
E-mail: Abder.Koukam@utbm.fr
*Corresponding author

Abstract: We study a local search approach for a coverage problem in the plane, called Balanced Honeycomb Clustering Problem (BHCP), where a honeycomb mesh is used to build irregular hexagonal clusters that have to cover a fixed amount of points of a given data distribution. This problem has application to dimension cellular networks adapted to radio-mobile traffic. The local search approach uses dynamic application of fitness landscape penalties in order to exit local minima and improve performances, while eliminating overloaded cells. Results are given in comparison to best solutions known generated by an evolutionary algorithm. For a considerable reduction of computational time, about 3000% lower, we show that local search outperforms a population-based evolutionary approach.

Keywords: cellular network; network dimensioning; adaptive mesh generation; evolutionary algorithms.

Reference to this paper should be made as follows: Créput, J-C. and Koukam, A. (2006) 'Local search study of honeycomb clustering problem for cellular planning', *Int. J. Mobile Network Design and Innovation*, Vol. 1, No. 2, pp.153–160.

Biographical notes: Jean-Charles Créput received his PhD in Computer Science from the University of Paris 6, France, in 1997. He is currently an Associate Professor in Computer Sciences and Engineering at the University of Technology of Belfort-Montbeliard, France, and performs research activity at the Systems and Transportation (SeT) Laboratory. His current research interest includes evolutionary algorithms and neural networks applied to telecommunications and intelligent transportation services.

Abderrafiaa Koukam is a Professor of Computer Science at the University of Technology of Belfort-Montbeliard, France. He received his PhD in Computer Science from University of Nancy I, France, in 1990. He heads research activities at Systems and Transportation (SeT) Laboratory on modelling and analysis of complex systems, including software engineering, multi-agent systems and optimisation.

1 Introduction

Honeycomb meshes are well known structures that are used to model cellular networks. Using a regular mesh, as explained by the theoretical cellular concepts (Walke, 2002), is the common representation of a cellular network. In Créput et al. (2005), it is proposed to dimension a network with possibly distorted hexagonal cells that cover the heterogeneous distribution of traffic load. Here, we extend and complete their results. Dimensioning is modelled using a geometric balanced clustering problem, called Balanced Honeycomb Clustering Problem (BHCP) in this paper, and referred as the Adaptive Meshing (AM) problem in earlier approach. A maximum capacity is fixed and characterised by a target traffic cost w^*, related to the maximum amount of simultaneous calls a Base Station (BS), represented by a single hexagonal cell, can handle. To cover the heterogeneous distribution of traffic load, hexagonal cells have to adapt their shapes with the main objectives of covering a traffic amount closest to the target cost and minimising the number of cells. Hence, geometrical constraints have to be satisfied to keep in mind the need for network regularity as explained by the theoretical cellular concepts.

While in other approaches (Calegari et al., 1997; Hurley, 2001; Vasquez and Hao, 2001; Zimmermann et al., 2003) of cellular planning, a graph formulation for the radio coverage problem is mentioned and related to a set covering problem (Eidenbenz et al., 1998; Feige, 1996), here we relate cell positioning to a geometric clustering problem in the plane. Analogous to Euclidean versions of standard problems, like k-median or TSP which have PTAS (Arora, 1998), we think that it leads to a more compact network representation and allows one to expect more rapid and efficient heuristics for it. Pragmatically, BHCP could be used as a first and rapid approximation step in order to help with the initial generation of candidate cells, avoiding the need for a wave propagation model in the initial step. In Walke (2002) the question of traffic load management is solved according to recursive decomposition of regular hexagonal structures. Here, the advantage is to allow the planner to design non-overlapping smooth transitions from low traffic density to high traffic density areas.

Cellular planning problems are generally NP-hard. We conjecture that BHCP is NP-hard. We think that the balanced clustering problem in the plane Planar X3C may be reduced to BHCP, using the same type of reduction as specified accordingly to Dyer and Frieze (1986), Lichtenstein (1982), Pferschy et al. (1994). Thus, to deal with large size problems, the use of meta-heuristic methods is encouraged. For example, evolutionary algorithms or tabu search were applied on the applications mentioned above. In Créput et al. (2005), BHCP is solved using an hybrid evolutionary algorithm. While local search quickly finds solutions in a small region of the search space, evolutionary operators determine interesting regions of the search space. It is a memetic algorithm (Moscato, 1999), often presented as a population-based algorithm incorporating a neighbourhood search heuristic, also called genetic local search (Mühlenbein, 1991). Here, we study local search, removing all other operators and working on a single solution. We take the benefit of problem knowledge issued from user experimenting interactive visual meshing. We show that, applying probabilistic fitness landscape penalties dynamically during local search process leads in a shorter time algorithm, which generates possibly superior quality results than the ones given by the hybrid evolutionary approach.

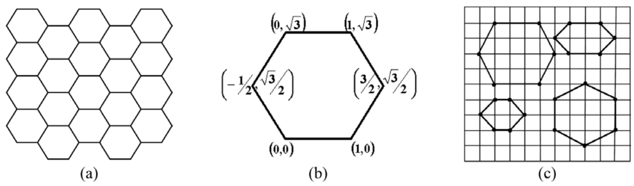
The paper is organised as follows. Section 2 presents the BHCP. Objectives and constraints are given. Section 3 presents the local search algorithm. Section 4 describes experiments with local search and Section 5 presents comparison with evolutionary approach. Finally, the last section is devoted to the conclusion and further research.

2 Problem statement

It is well known that there are three possible tessellations of a plane with regular polygons of the same kind: square, triangular and hexagonal, corresponding to dividing a plane into regular squares, triangles and hexagons, respectively (Bern and Eppstein, 1995). The hexagonal tessellation is called a honeycomb mesh (Stojmenovic, 1997; Zhang and Wang, 2002).

One way to define a honeycomb mesh is to build it from basic hexagons inside a rectangle as in Figure 1(a) yielding to a 5×4 honeycomb rectangular mesh. The unit hexagon with radius 1 is shown in Figure 1(b). It is a perfect, or regular, hexagon encoded necessarily as algebraic number coordinates. A perfect, or regular, mesh is constituted of perfect hexagons. The size of a perfect hexagon is either the length of an edge or the radius of the circle, which embeds vertices. Figure 1(c) shows approximate hexagons embedded into a finite precision grid. Following the cellular network metaphor, hexagons are cells and clusters are covering cells where to install Bases Stations. Given a point p in the plane, we say that an hexagon covers p if the point lies on the interior of the hexagon. Here, it is assumed that a value, called a weight, is associated to a point p, representing mobile traffic demand at this point, for example given in Erlang. We define the load of a cell, or its weight, as the sum of the weights of its covered points.

Figure 1 (a) A rectangular honeycomb mesh, (b) a single perfect hexagon defined by its 6 vertex coordinates and (c) approximate hexagons into a finite grid



2.1 Balanced honeycomb clustering problem

Given a finite set P of weighted points in the plane, a maximum capacity value w^* of a cell, and given the size $N \times M$ of a honeycomb rectangular mesh, the problem consists in finding hexagon vertex coordinates so that the following objectives are minimised:

1. *Nvis*: number of visible cells, that is, those covering some traffic
2. *Dev*: adaptation of visible cells, defined by
$$\text{Dev} = \frac{100 \times \sum_{i=1}^{\text{Nvis}} \left| w_i - w^* \right|}{\text{Nvis} \times w^*} \quad (1)$$
where w_i is the cell weight.
3. *Geom*: geometry measurement of visible cells (distortion measure), defined by
$$\text{Geom} = \frac{100 \times \sum_{i=1}^{\text{Nvis}} \left(\frac{1}{12} \sum_{j=1}^{6} \left(\frac{\alpha_{ij} - 120}{60} \right)^2 + \frac{1}{12} \sum_{j=1}^{6} \left(\frac{d_{ij} - \bar{d}_i}{\bar{d}_i} \right)^2 \right)}{\text{Nvis}} \quad (2)$$
where α_{ij} (respectively, d_{ij}) denote an angle in degrees (respectively, an edge size), $1 \le j \le 6$, and $\bar{d}_i$ is the mean of the hexagon edge lengths.
4. *Nov*: number of overloaded cells, that is, those exceeding capacity w^*.
5. *Dov*: adaptation of overloaded cells, defined by
$$\text{Dov} = \frac{100 \times \sum_{i=1}^{Nov} \left(w_i - w^* \right)}{\text{Nov} \times w^*} \quad (3)$$
6. *Max*: maximum traffic coverage for a single cell exceeding capacity w*.

The goal is to generate an adapted mesh. Starting from a perfect honeycomb rectangular mesh, cells are subject to deformations in such a way that each cell has to cover a target of traffic load defined by the target weight w^*, with respect to geometric constraints on shape and topology. On a radio system, w^* is the target capacity of a BS. Here, we assume that w^* is an input, it is fixed and remains constant. In practice, the point set P, called traffic map, is bit-mapped on a fine-grained map of size $m \times n$, delimiting some geographic area. Thus, each traffic point, called a pixel, has integer coordinates into the underlying map. A pixel colouring algorithm, as Bresenham algorithm, is used to determine which traffic points are covered by a cell and to compute cell load.

The traffic map is composed either of regions containing traffic or regions without traffic. We then distinguish a set of visible or useful cells that cover some traffic (non-null load), and a set of invisible cells that do not cover traffic (null load). Invisible cells are not involved in the resulting mesh, as none BS will be associated to them. But these cells are necessary in the optimisation process while any cell may move from null load to non-null load status and vice versa. The size $N \times M$ of the honeycomb mesh is an input. Thus, starting with a uniform mesh and due to the fact that traffic is heterogeneous, the number of visible cells will be slightly increased or decreased. We don't specify an upper physical limit for a given cell. Its size is only constrained by traffic adaptation and by its surrounding cells inside the enclosing fine-grained map delimiting the geographic area.

We can note that modelling network dimensioning by BHCP has some disadvantages since cells may not map to site candidate transmission site locations and since geographic particularities of the terrain are not considered using a wave propagation model. Generally, BS parameters settings on real environment never lead to regular honeycomb architecture. Real cell coverage is always self-adapted to mobile traffic heterogeneity and to propagation variation, ending to different sizes and shapes of cells. Then, there is always a high gap between the cost and the quality of regular theoretical networks and irregular real networks. Here, we place the emphasis on reducing such a gap by generation of irregular cells adapted to traffic heterogeneity, at an initial step in network design.

The main difference with other approaches of cellular planning, such as the ones presented in Calegari et al. (1997), Hurley (2001), Vasquez and Hao (2001), Zimmermann et al. (2003), is that in our approach antenna location is modelled using geometric entities in plane rather than using graphs. According to decomposition of the overall planning process, we take the assumption that this preliminary step has to be as short as possible, with relatively low computational cost and no need for a detailed wave propagation model. We think that BHCP has to be used as a model that encompasses implicitly some real world issues, in a complementary way to the ones developed for the design step. For example, frequency reuse is supposed to be achieved subsequently with standard algorithms on honeycomb cells as soon as the hexagonal topology is preserved. Overlapping between cells is not considered since there are no overlapping cells in the honeycomb mesh by definition. There is no territory shading since it is supposed to be flat and hence the propagation uniform. Robustness under traffic growth will be achieved by adaptation of covering areas, get from antenna parameters settings.

The goal of experiments will be to eliminate overloaded cells completely, transforming last three objectives: Nov, Dov, Max into the constraint Nov = 0, Dov = 0 or Max $\leq w^*$. The Dev and Dov traffic adaptation criteria are normalised mean deviation to w^* rather than squared deviation. It is because mean deviation has a more natural meaning. Whereas, we will see that traffic adaptation addresses a compromise between reduction of variance and mean deviation. The Geom criterion evaluates a normalised distortion between cells and a regular hexagon.

3 Local search algorithm

3.1 Algorithm

The local search is an adaptive process that applies mutations on isolated vertices with a basic selection mechanism that make an individual, which is a single solution, evolves to a local minimum. The mutation operator, called micro-mutation, performs a little move of some randomly chosen vertex inside an inner area as illustrated in Figure 2. The size of the area defines the mutation intensity. This local perturbation authorises a fast incremental fitness computation. The local search algorithm with its default parameters is given in Figure 3. The selection validates the move depending if it improves or not the fitness.

Figure 2 Local search mutation

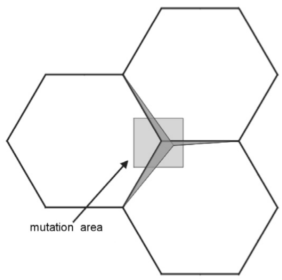


Figure 3 Local search algorithm

```
Initialise a single solution with a uniform honeycomb mesh
For n ∈ {1, …, Gen} // Perform Gen generations (Gen = 1000 in experiments)
        i ← M (mutation intensity, M = 6 pixels by square side in practice)
        c ← C (number of times the mutation is allowed to fail, C = 1000 in practice)
        t ← T (maximum number of iterations in one generation, T = 5000 in practice)
        While i ≠ 0 and t > 0 do
                t ← t - 1
                // individual fitness is f
                micro-mutate with intensity i
                evaluate new fitness f' (this is done by a local evaluation)
                if f' > f then
                        undo micro-mutation
                        c ← c - 1
                        if c = 0 then
                                i ← i - 1 // decrease intensity
                                c ← C
        end While
        n ← n + 1 // next generation
end For
```

3.2 Fitness function

The fitness function is a scalar function. It evaluates the quality of solutions during the search process. The fitness function is constructed so as to deal efficiently with the objectives. Three important terms are summed using different weighting coefficients. They correspond, respectively, to traffic adaptation between cells, respect of geometrical constraints and minimisation of the number of visible cells. Elimination of overloaded cells is expressed into the first aggregating term. The fitness function is thus defined as in

$$F = k_A f_A + k_G f_G + k_N f_N \quad (4)$$

where f_A is the evaluation of resource adaptation, f_G is the evaluation of geometrical constraints, f_N the evaluation of visible cells number and k_A, k_G, k_N the weighting coefficients.

Adaptation of the whole mesh is defined as the sum of individual cell adaptation values divided, for normalisation purposes, by the total number of cells. It is given by

$$f_A = \frac{\sum_{i=1}^{N} a_i}{N} \quad (5)$$

where N is the total number of cells in the rectangular honeycomb mesh.

To locally adapt border of cells to traffic frontier, the fitness aggregates penalties depending on the proportion z_i of surface of a cell i outside a traffic region, that is, the number of pixels with zero value divided by the total number of pixels covered by the cell. As illustrated in Figure 4(a), four cases are relevant for a cell i with weight w_i:

1 The cell is completely outside the traffic area R as is m_0 where $z_i = 1$. It is an invisible cell and no BS is associated too. Its adaptation value a_i is then

$$a_i = 0 \quad (6)$$

2 The cell is almost completely inside or totally inside a traffic region, that is, $z_i < 0.1$, m_3 and m_4 alike. The adaptation value is stated in

$$a_i = h\left(\frac{w_i - w^*}{w^*}\right) \quad (7)$$

where h is a filtering function that controls and penalises deviation of a single cell.

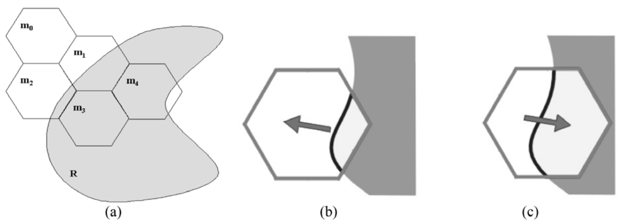
3 The cell is almost outside the traffic area, that is, $z_i > 0.5$ and $w_i/w^* < 0.5$, as is m_2. It is the case shown in Figure 4(b). The cell is also an invisible cell and it has to exit from the traffic region with a penalising term, as in

$$a_i = 10 \times (1 - z_i)^2 \quad (8)$$

4 The cell is into an intermediate state, that is, $z_i \geq 0.1$, like m_1. It is the case illustrated in Figure 4(c). In this case, the cell is made to enter the traffic region by adding a specific weight to its deviation value. This is described in

$$a_i = h\left(\frac{w_i - w^*}{w^*}\right) + z_i^2 \quad (9)$$

Figure 4 Different cases of cell covering



The geometrical constraint imposes a cell to have a shape closest to a regular hexagon. The function takes into account the angles of the hexagon, which must approximate 120°, and sizes of its edges, which should be of equal length. The f_G function is as follows:

$$f_G = \frac{\sum_{i=1}^{N} g_i}{N} \quad (10)$$

where the geometrical evaluation g_i of a single cell is given by

$$g_i = \frac{1}{12}\sum_{j=1}^{6}\left(\frac{\alpha_j - 120}{60}\right)^2 + \frac{1}{12}\sum_{j=1}^{6}\left(\frac{d_j - \bar{d}}{\bar{d}}\right)^2 \quad (11)$$

where α_j denotes angle j in degrees of cell i, d_j denotes the size of edge j of cell i and $\bar{d}$ is the mean size of edges of cell i.

The term f_N of the objective function is the number of visible cells divided by the total number of cells in the whole mesh. It is defined by

$$f_N = \frac{N_v}{N} \quad (12)$$

where N_v the number of visible cells.

The different fitness sub-terms f_A, f_G, f_N are normalised according to, respectively, the ideal weight w^*, an ideal regular hexagon and the total number of cells. Then, metric of fitness is defined by these ideal values that put sub-term values in comparable ranges. An important parameter is the filtering function h of the deviation between weight w and capacity w^*. Experiments will study its impact on solution quality.

4 Local search study

4.1 Filtering functions and goal of experiments

An important term of the fitness function (4) is the traffic adaptation term (5) which incorporates a filtering function h, used in (7) and (9). Since one expects to reduce deviation and variance to the ideal cost w^*, and eliminate overloaded cells, choice of function h has to be made carefully. The filtering function h can be specified so as to penalise more or less deviation between the weight w and the capacity w^*. Depending on its shape, it will penalise outliers and overloaded cells and favour emergence of non-overloaded cells with the weight closest to the target cost w^*. In the following experiments, function h will be instantiated in different ways and their impact evaluated separately. Then, a probabilistic fitness combination mechanism will be applied, using the best performing instantiation cases, to improve solution quality by dynamically exiting local minima.

To reduce deviation to the ideal cost w^*, natural choices of function h would be the absolute value function $h_0(x) = |x|$ or the quadratic function $h_1(x) = x^2$. Considering population of cells and as it is known in statistics, function h_0 has a natural meaning as a mean deviation, however function h_1 addresses reduction of variance rather than mean deviation. Since function h_1 is more sensitive to outliers, it follows that it could be more efficient into the local search process. Function h_1 could lead to better compromises between reduction of variance together with reduction of mean deviation to w^*. Mean deviation to w^* is regarded, however, as an easy to interpret objective of the problem. Function h is instantiated with two types of shapes. The first type of instantiation uses symmetric functions. The second type of instantiation uses asymmetric functions that penalise overloaded cells with the goal to eliminate them from the mesh. Six different filtering functions are used. They are visualised in Figure 5 and defined by

$$h_0(x) = |x| \tag{13}$$

$$h_1(x) = x^2 \tag{14}$$

$$h_2(x) = |x|^3 \tag{15}$$

$$h_3(x) = \exp(|x|) - 1 \tag{16}$$

$$h_4(x) = \begin{cases} \log(9x+1) \text{ if } x > 0 \\ x^2 \quad \text{otherwise} \end{cases} \tag{17}$$

$$h_5(x) = \begin{cases} (x+2)^3 - 7 \text{ if } x > 0 \\ x^2 \quad \text{otherwise} \end{cases} \tag{18}$$

4.2 Filtering functions in local search algorithm

The impact of the filtering function choice was evaluated by running 3 runs, with 1000 generations by run, in each case and selecting the best result. Coefficients of fitness function sub-terms k_A, k_N, presented in (4), were set to value 1, and k_G to values 1, 2 and 3 to modulate importance of geometry. The test case is the data distribution presented in Figure 8(a). It presents an elliptic field, with low traffic values, inside of which an A shape is written with density values five times greater. Results are summarised in Table 1 and two typical results are drawn in Figure 6 to illustrate opposite effects induced by filtering function. In the first case (a) with function h_1, the mesh falls into a local minima with distorted cells, but showing a good compromise between objectives. In the second case (b), using asymmetric function h_4 leads to a curious effect where overloaded cells growth and reduce their number, living the designer free to refine again the delimitated zones.

Table 1 Comparative results using different filtering functions

Filtering function	k_G	*Dev*	*Geom*	*Nvis*	*Nov*	*Dov*	*Max*
h_0	1	32.66	12.57	451	121	47.42	8980
h_0	3	34.90	7.34	451	116	52.19	8680
h_0	6	37.97	5.01	451	121	55.32	9320
h_1	1	10.56	11.73	451	125	7.49	3920
h_1	3	18.97	7.47	451	159	15.12	4540
h_1	6	27.18	5.40	451	172	24.01	5280
h_2	1	18.48	9.12	451	158	15.18	4120
h_2	3	28.84	5.57	451	170	26.58	4660
h_2	6	35.18	4.00	451	169	34.81	4800
h_3	1	19.61	18.11	451	127	22.27	5460
h_3	3	21.34	9.66	451	142	21.30	5340
h_3	6	24.88	7.10	451	152	24.58	6000
h_4	1	31.97	17.02	451	11	563.80	40,180
h_4	3	46.35	9.88	450	24	378.62	38,100
h_4	6	49.91	6.22	451	31	311.15	27,020
h_5	1	21.04	18.81	451	63	53.91	5980
h_5	3	19.07	15.26	451	71	35.91	5100
h_5	6	29.12	11.98	451	94	48.68	5560

Figure 5 Filtering functions h_0, h_1, h_2, h_3, h_4, h_5

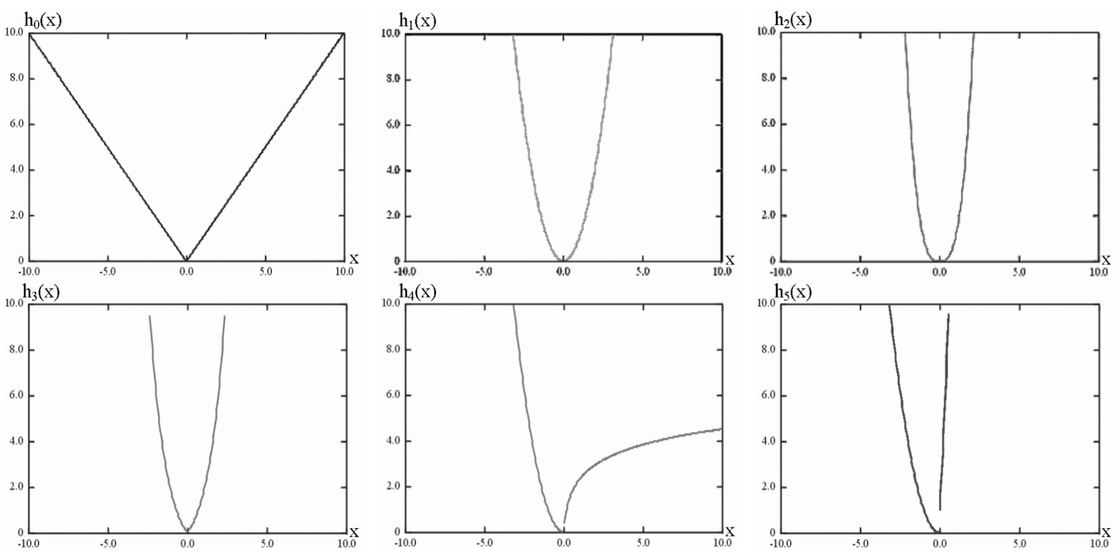
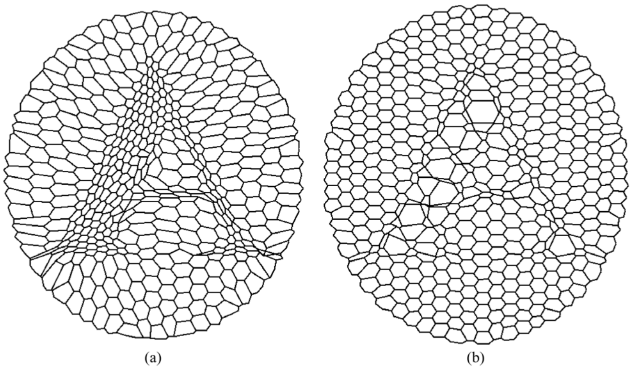


Figure 6 (a) With filtering function h_1 and (b) with asymmetric filtering function h_4



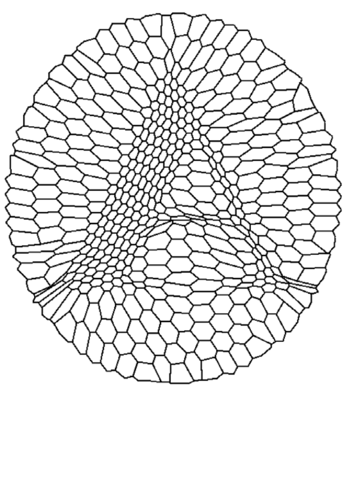
4.3 Probabilistic aggregative fitness function

To exit local minima and try to eliminate overloaded cells, one way is to introduce new operators and build an evolutionary approach as in Créput et al. (2005). Here, we maintain on local search with a single individual and simply modify fitness landscape dynamically using a probabilistic application of one of the filtering functions $h_{0\text{–}5}$, at each generation. This mechanism is similar to probabilistic fitness combination in multi-objective methods (Coello Coello, 1999). Following preliminary tests, the probabilities chosen were 0.45, 0.40, 0.10, 0.05 for functions h_1, h_2, h_3 and h_4, respectively. We call it probabilistic fitness $g(x)$. Fitness function coefficients k_A, k_N, k_G, presented in (4), were set to value 1. Typical results obtained within a single run are given in Table 2, and one of the meshes obtained is drawn in Figure 7. The result can be compared to the h_1 case of Figure 6(a). For the same number of cells and same iteration number, it shows a clear improvement on the main objectives of mesh adaptation and overloaded cell elimination, while maintaining geometry in quite a better regularity.

Table 2 Simulation results using probabilistic aggregative fitness function

Fitness	*Dev*	*Geom*	*Nvis*	*Nov*	*Dov*	*Max*
g(x)	6.33	11.75	451	20	1.29	3020
g(x)	5.99	11.28	451	32	1.40	3000
g(x)	6.23	11.09	451	42	2.25	3108
g(x)	7.04	10.72	451	66	3.41	3164

Figure 7 Mesh obtained using probabilistic aggregative fitness $g(x)$



5 Comparison with evolutionary approach

Here, we use a probabilistic fitness, $g'(x)$, with asymmetric filtering function h_5, with a threshold, rather than h_4, to completely eliminate overloaded cells and try to outperform results from evolutionary approach of Créput et al. (2005). After a first round of experiments, probabilities of filter function application were chosen as 0.45, 0.05, 0.15, 0.40 for functions h_1, h_2, h_3 and h_5, respectively, at each generation. The parameters defining a simulation are given in Table 3. The size $N \times M$ of the honeycomb mesh and the size of the underlying traffic demand map $n \times m$ are chosen prior to optimisation to be closest to the ideal number of cells $N^* = S/w^*$, where S is the total sum of traffic in the map. The parameter h is the probabilistic function $g'(x)$ in local search approach, or the single h_5 in the evolutionary approach. Other input parameters are the coefficients k_A, k_G and k_N of the aggregative fitness function, the population size *Pop*, which is 1 in local search, and the number of generations *Gen* that is fixed to 1000.

Table 3 Simulation parameters

Parameter	*Significance*	*'A' test case*	*'France' test case*
w^*	Ideal traffic covering cost	2900	80,000
S	Sum of traffic in the map	119.10^6	116.10^7
N^*	Ideal number of cells $N^* = S/w^*$	412	145
$n \times m$	Size of the traffic map	600×600	600×600
$N \times M$	Size of the honeycomb mesh	30×30, 31×30, 31×31	23×24, 24×24
k_A	Fitness adaptation coefficient	6	6
k_G	Fitness geometry coefficient	1;3	1;3
k_N	Fitness visible cells coefficient	1	1
h	Deviation function h_5 or g	g'(0.45, 0.05, 0.15, 0.40)	g'(0.45, 0.05, 0.15, 0.40)
Gen	Number of generations	1000	1000
Pop	Population size	1	1

Two input instances, 'A' and 'France' test cases, were considered with varying characteristics of size and traffic as given in Table 3, and drawn in Figures 8(a) and 9(a) through sample points extracted from density map using a roulette wheel mechanism. Simulations were done three times per case using parameters in Table 3. Since we took 1000 generations as in the evolutionary approach, local search simulation time was, proportionally to population size, around 30 or 40 times shorter. Local search took approximately 20 min on a Sun Workstation 750 MHz.

Figure 8 (a) Density sampling, (b) and meshing for 'A' test case, without and (c) with surrounding invisible cells

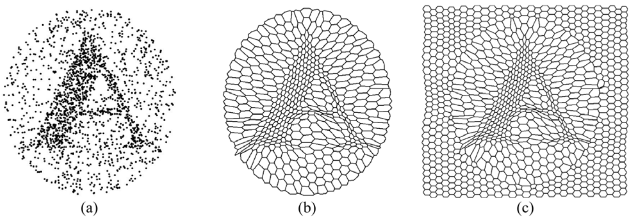
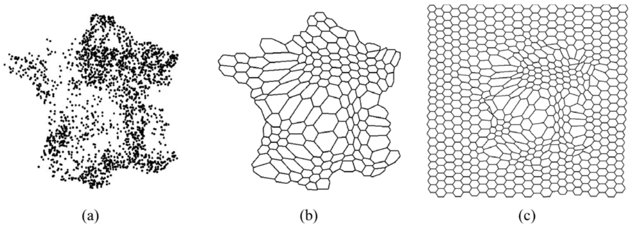


Figure 9 (a) Density sampling, (b) and meshing for 'France' test case, without and (c) with invisible cells



Best results are reported in Table 4, together with the ones obtained by the evolutionary approach. Typical results are drawn in Figures 8 and 9 with (c) without (b) surrounding cells. They illustrate the ability of the approach to produce a well contoured mesh on a map while eliminating overloaded cells. As given in Table 4 for the two test cases, the constraint on overloaded cells being satisfied, local search with probabilistic fitness is able to generate better solutions than evolutionary approach does, considering the three objectives Dev, Geom and Nvis simultaneously.

Table 4 Comparison of local search with evolutionary approach (CKLC05)

Method	*Instance -k_A-k_G-k_N-h-Pop*	*Dev*	*Geom*	*Nvis*	*Nov*	*Dov*	*Max*
Local search (probabilistic fitness)	A-6-3-1-g'-1	11.59	9.44	482	0	0	2896
	A-6-3-1-g'-1	9.10	9.89	467	0	0	2900
Evolutionary approach (CKLC05)	A-6-3-1-h_5-30	14.77	9.63	499	0	0	2894
	A-6-3-3-h_5-30	13.58	10.40	494	0	0	2896
	A-6-3-6-h_5-30	10.63	11.35	473	1	4.14	3020
	A-6-3-9-h_5-30	11.77	10.30	478	1	0.24	2907
Local search (probabilistic fitness)	France-6-3-1-g'-1	11.87	7.52	169	0	0	79,995
	France-6-3-1-g'-1	7.74	8.40	161	0	0	79,979
Evolutionary approach (CKLC05)	France-6-3-1-h_5-40	12.09	7.65	169	0	0	79,899

Generally, it is admitted that evolutionary algorithms are relatively slow, but that higher global optima may be attainable over a much longer time period. This point is illustrated here considering number of visible cells (with non-null load). For example, the number of visible cells slightly increases or decreases during evolutionary search, by introducing new cells on traffic regions and leading to a wider search space exploration. Using local search, the number of visible cells remains constant during the search and depends on the initial mesh size $N \times M$, thus reducing the search space. Nevertheless, probabilistic fitness mechanism significantly improves local search performance, working with a constant number of useful cells with relatively short computation time.

6 Conclusion

This paper addressed a local search approach to the mobile network dimensioning problem. It transferred the dimensioning problem to a geometric meshing generation problem that encompasses traffic requirements, minimisation of number of cells and preservation of structured hexagonal topology. This problem is closely related to usual balanced clustering problems in the plane. We presented a solution to this new optimisation problem through a local search heuristic, using probabilistic fitness aggregation with variable and dynamic penalties to help exit from local minima. Experiments illustrate the potentiality of the method at reaching better solutions than existing evolutionary approach does for the same problem. The improvement leads to a reduction of the number of transmitters and the production of irregular cell respecting hexagonal and neighbourhood constraints from theoretical cellular concepts. The next step will consist in relating ideal meshes found by the approach to the physical antennae parameters and using a wave propagation model.

Acknowledgement

We acknowledge the help of J.N. Martin from the University of Technology of Belfort-Montbeliard and of all members of the network optimisation team of France Telecom R&D at Belfort, France.

References

Arora, S. (1998) 'The approximability of NP-hard problems', *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, STOC'98, pp.337–348.

Bern, M. and Eppstein, D. (1995) 'Mesh generation and optimal triangulation', in D-Z. Du and F.K. Hwang (Eds). *Computing in Euclidean Geometry*, 2nd editon, World Scientific, pp.47–123.

Calegari, P., Guidec, F., Kuonen, P. and Kobler, D. (1997) 'Parallel island-based genetic algorithm for radio network design', *Journal of Parallel and Distributed Computing*, No. 47, pp.86–90.

Coello Coello, C.A. (1999) 'A comprehensive survey of evolutionary-based multiobjective optimization techniques', *Knowledge and Information Systems, An International Journal*, Vol. 1, No. 3, pp.269–308.

Créput, J.C., Koukam, A., Lissajoux, T. and Caminada, A. (2005) ‘Automatic mesh generation for mobile network dimensioning using evolutionary approach’, *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 1, pp.18–30.

Dyer, M.E. and Frieze, A.M. (1986) ‘Planar 3DM is NP-complete’, *Journal of Algorithms*, Vol. 7, pp.174–184.

Eidenbenz, S., Stamm, C. and Widmayer, P. (1998) ‘Positioning guard at fixed height above a terrain – an optimum inapproximability result’, *Lecture Notes in Computer Science*, Vol. 1461, pp.187–198.

Feige, U. (1996) ‘A threshold of ln n for approximating set cover’, *Proceedings of the 28th Annual Symposium on the Theory of Computing, ACM*, pp.314–318.

Hurley, S. (2001) ‘Planning effective cellular mobile radio networks’, *IEEE Transactions on Vehicular Technology*, Vol. 51, No. 2, pp.443–472.

Lichtenstein, D. (1982) ‘Planar formulae and their uses’, *SIAM Journal on Computing*, Vol. 11, No. 2, pp.329–343.

Moscato, P. (1999) ‘Memetic algorithms: a short introduction’, in D. Corne, M. Dorigo and F. Glover (Eds). *New Ideas in Optimization*, McGraw Hill.

Mühlenbein, H. (1991) ‘Evolution in time and space – the parallel genetic algorithm’, in G. Rawlins (Ed). *Foundations of Genetic Algorithms*, Los Altos, CA: Morgan Kaufmann.

Pferschy, U., Rudolf, R. and Woeginger, G.J. (1994) ‘Some geometric clustering problems’, *Nordic Journal of Computing*, Vol. 1, No. 2, pp.246–263.

Stojmenovic, I. (1997) ‘Honeycomb networks topological properties and communication algorithms’, *IEEE Transactions On Parallel and Distributed Systems*, Vol. 8, No. 10, pp.1036–1041.

Vasquez, M. and Hao, J.K. (2001) ‘A heuristic approach for antenna positioning in cellular networks’, *Journal of Heuristics*, Vol.7, pp.443–472.

Walke, B.H. (2002) *Mobile Radio Networks*, John Wiley.

Zhang, H. and Wang, G. (2002) ‘Honeycomb subdivision’, *Journal of Software*, Vol. 14, No. 4.

Zimmermann, J., Hons, R. and Muhlenbein, H. (2003) ‘ENCON: evolutionary algorithm for the antenna placement problem’, *Computers and Industrial Engineering*, Vol. 44, pp.209–226.