Optimising base station location for UMTS cellular networks

To cite this article: G Kalata et al 2014 J. Phys.: Conf. Ser. 570 062005

View the article online for updates and enhancements.

Related content

- Implementation of a Fractional Model-Based Fault Detection Algorithm into a PLC Controller
  Ryszard Kopka

- Analysis of Usefulness of a Fuzzy Transform for Industrial Data Compression
  Anna Sztyber

- Properties of input-output Hammerstein-bilinear structure with application to an industrial air handling unit
  T Larkowski, I Zajic, K J Burnham et al.
Optimising base station location for UMTS cellular networks

G Kalata, I Pozniak-Koszalka, L Koszalka and A Kasprzak
Department of Systems and Computer Networks
Wroclaw University of Technology, Wroclaw, Poland

E-mail: leszek.koszalka@pwr.edu.pl

Abstract. Rapid development of universal mobile telecommunication systems put demands on tools for assisting planning of cellular network infrastructure. The tools need to focus on critical issues in modern cellular networks and techniques used for previous generation system no longer serve useful. In this paper, an algorithm based on Branch & Bound approach is proposed for solving base station location problem, covering interference levels, traffic demands and power control mechanism. The efficiency of the algorithm is evaluated with respect to existing approaches for solving this problem – using the designed and implemented experimentation system.

1. Introduction
The problem of choosing base station location is one of the basic problems that arise in 3G network planning, such as Universal Mobile Telecommunication System (UMTS). Scarcity of radio resources and high cost of infrastructure and resources put high demands on the quality of network planning. Since 3G systems are based on wideband code-division multiple access (W-CDMA), all connections share the same bandwidth. The capacity of the system depends on achievable signal-to-interference ratio (SIR) values. Because 3G systems tend to be very sensitive to interference, it should be ensured that any base station (BS) does not generate or receive too much of it [1]. Thus, the power control techniques are employed. This mechanism ensures that each device in the network transmits and receives just enough power to allow communication [2]. The mobile stations emission power is limited; therefore a minimal SIR level needed for communication may be unreachable in presence of high interference. Also, because traffic distribution can vary with time (and with them the interference levels), the area covered by a single BS can vary significantly. Usually, in planning phase of cellular network, a set of candidate sites for base stations is considered. Also traffic distribution needs to be supplied; usually it is estimated using empirical prediction models. In this work, the problem defined in [3] is taken into consideration. The model focuses on up-link direction (i.e. mobile station to BS), quality requirements in terms of SIR minimal levels and power limits.

The rest of work is organised as follows. In section 2 the problem is formulated. In section 3 two known algorithms based on randomized greedy procedures and Tabu Search are described. Then, in section 4 the algorithm based on Branch & Bound approach is presented. Section 5 and section 6 describe the created simulation environment and the propagation model, respectively. In section 7 the designed experiment setup is considered and the obtained computational results are presented. The conclusions appear in section 8.

1 Corresponding author.
2. Problem statement

The problem of the base station location may be interpreted as finding such a subset of a given set of base station candidate sites \( S \), which results in the lowest cost and lowest minimal power of transceivers. The solution must cover all given test points and guarantee at least minimal connection quality. The problem can be formulated (basing on [3]) as follows:

Given

- set of candidate sites (CS) \( S = \{1, ..., m\} \) with corresponding installation cost \( c_j, j \in S \),
- set of test points (TP), \( I = \{1, ..., n\} \) with corresponding required number of simultaneously active connections \( u_i, i \in I \),
- propagation gain matrix 
  \[
  G = \{g_{ij}\}_{1 \leq i \leq n, 1 \leq j \leq m}, 0 < g_{ij} \leq 1,
  \]

To find

- base station design \( Y = \{y_j\}, j \in S \),
  \[
  y_j = \begin{cases} 
  1 & \text{if a BS is installed in} \\
  0 & \text{otherwise}
  \end{cases},
  \]
- test points assignment \( X = \{x_{ij}\}, i \in I, j \in S \),
  \[
  x_{ij} = \begin{cases} 
  1 & \text{test point } i \text{ is assigned to BS } j \\
  0 & \text{otherwise}
  \end{cases},
  \]

Such that

\[
\sum_{j=1}^{m} c_j y_j + \lambda \sum_{i=1}^{n} \sum_{j=1}^{m} u_i \frac{1}{g_{ij}} x_{ij} \text{ is minimal} \tag{1}
\]

Subject to the constraints

\[
\sum_{j=1}^{m} x_{ij} = 1, i \in I \tag{2}
\]

\[
x_{ij} \leq \min \left\{ 1, \frac{g_{ij} p_{\text{max}}}{p_{\text{target}}} \right\}, i \in I, j \in S \tag{3}
\]

\[
y_j \left( \sum_{h=1}^{n} \sum_{t=1}^{m} u_t \frac{g_{ht}}{g_{ht}} x_{ht} - 1 \right) \leq \frac{1}{\text{SIR}_{\text{min}}}, j \in S \tag{4}
\]

The first term of cost function (1) refers to the total cost of base station design, the second one to the total needed power. The constraint (2) implies that all test points should be covered, the constraint (3) ensures that TP can be assigned only to BS with enough signal strength, and the constraint (4) implies that for each BS used, the signal-to-interference ratio (SIR) is high enough to allow communication. As can be seen, after choosing candidate sites, one must also assign each TP to one of the available BS. Solution to this assignment sub-problem is trivial. As was shown in [3] each traffic point \( i \) has to be assigned to \( j \)-th BS with the highest propagation gain \( g_{ij} \). The problem is NP-hard.
3. Related work – Reference algorithms
To make some reference, two greedy randomized procedures and Tabu Search algorithm were adopted from [3]. In following subsection this algorithm will be described. The common procedure used in all three algorithms is the construction of \( X \) matrix. Because all of these algorithms can generate intermediate solutions, that are unfeasible. As has been stated in Section 2, each \( i \)-th TP should be assigned to available \( j \)-th BS that has the highest \( g_{ij} \) to ensure lowest interference and power demands. So, the \( X \) matrix construction procedure first assigns each TP to BS according to above rule. For every base station SIR is computed, and if it is lower than SIR\( _{\text{min}} \), then all test points associated with it are considered for elimination. Next, the test point with the highest emission power is unassigned and the procedure repeats, until all BS fulfils SIR demands. Due to elimination of test point that cannot be handled, one can infer how much of the traffic can be served using current subset of BS. The drawback of this approach is that it quickly becomes computationally expensive.

3.1. Randomized greedy add procedure
This procedure starts from empty set of BS and in each iteration evaluates set with added BS from remaining candidate sites. The candidate sites are scored according to greedy function which captures the current total cost and the amount of currently covered traffic. This greedy function incorporates trade-off parameter between cost and amount of covered traffic. Then, from few best candidates according to greedy function one is chosen randomly, and added to the current solution. This procedure is repeated, until adding new base station no longer improves the greedy function.

3.2. Randomized greedy remove procedure
This procedure is similar to the previous one. It starts from an initial solution consisting of all the base stations, and then iteratively removes BS according to greedy function in analogous manner that previous procedure. However, the greedy function is somewhat different. It includes an additional term, which measures the number of additional connections that could be served by the solution. The advantage of this procedure is that, if a feasible solution exists, it always finds it - since solution including all candidates must fulfil the constraints (if instance is solvable).

3.3. Tabu Search
The Tabu Search algorithm [4], [5] uses swap moves and greedy goal function from randomized greedy remove procedure. The swap moves remove from the solution currently included BS and add one BS from currently unused sites. Because the number of all possible swap moves is too large to enumerate and evaluate them all, only some “best” candidates for each currently used BS are taken into consideration. The “best” refers to candidates that have higher propagation gains with respect to currently used BS. Some fraction of “best” candidates are systematically evaluated, the other are evaluated with a given probability. The algorithm was used in multi-run configuration, that is, the algorithm was invoked 10 times, and the best solution from 10 runs was returned. The initial solution for each run was generated using one of the randomized greedy procedures, and then 200 iterations of Tabu Search was computed.

4. Branch & Bound algorithm
The Branch & Bound is a general algorithm used for solving optimisation problems. The method was first proposed in [6] and widely developed [7]. The algorithm enumerates possible solutions by means of solution tree traversal. It uses lower bound and other criteria to eliminate branches of solution tree that represents fruitless solutions. In [4] Branch &Bound algorithm was used to solve differently modelled problem of base station optimisation, which is also NP-hard. They employed greedy solution tree traversing to ensure that optimal solution is found as early as possible. For smaller instances the algorithm could be used to provide the exact solution, and for larger it can be simply turn into heuristic by limiting the number of iterations. This concept was adopted into this paper.
In the beginning of the activity of the proposed algorithm, for each $i$-th TP, a list of feasible BS is created. This list is then sorted using signal strength $g_{ij}$ from higher to lower. Additionally, the BS candidate sites are sorted according to their scores, which are evaluated as follows. For each $i$-th TP the score of every feasible BS that this TP can be assigned to, is increased by 1. Thus, BS that can cover many test points will be placed on the beginning of the list. These two kinds of lists are used in the next actions of the algorithm.

4.1. Initial solution

At first, an initial solution is evaluated in order to get the upper bound of optimal solution's objective function. The lower bound for the number of BS needed to serve all traffic points is evaluated. It is easily to show that the maximum number of TP that can be served by any BS, while fulfilling demand for SIR, can be computed using the equation (5):

$$\max_{TP} = \left\lfloor \frac{1}{\text{SIR}_{\text{min}}} + 1 \right\rfloor$$

what implies that for $n$ TP the lower bound of number of BS $\min_{BS}$ can be expressed by (6):

$$\min_{BS} = \left\lfloor \frac{n}{\max_{TP}} \right\rfloor$$

Thus, the first best $\min_{BS}$ BS is included into initial solution. Then, every TP is assigned to best available BS (using per TP list). After that in any iteration one of the remaining BS is added, and the solution is checked for feasibility and objective function evaluated. If solution is feasible, and it is better (according to objective function defined by (1)) than a previous solution, it becomes current initial solution and the next BS is added. If current solution is not better than a previous one, the initial solution deriving procedure is interrupted.

4.2. Branching

The solution tree is constructed in the following way. At each node of the tree the next BS from best BS list (evaluated at the beginning) is taken into consideration, and two new nodes produced. One is considered to not use this BS, and the other to do so. For each of the nodes the lower bound is evaluated. If lower bound is higher than current upper bound, the node is removed. Remaining nodes are checked for feasibility, and if they pass they are added to active nodes priority queue. The priority of the nodes is its greedy score. In any iteration of Branch & Bound algorithm, one node of the solution tree is popped out from the queue. Its lower bound is compared with current upper bound (which could change meanwhile) and further expanded, or if it is at maximum depth of the tree it is evaluated using objective function. If such solution is feasible and has the lowest objective function value so far, it becomes the best solution and sets the new upper bound. The algorithm ends after given number of iterations (acts like heuristic) or when the whole tree is traversed (the obtained solution is known to be optimal).

4.3. Lower bound of objective function

In order to compute lower bound of the objective function the one must assign each TP to some of the available BS. Because only a part of solution is known at that time, the following strategy is used. For each TP, the BSs from feasible list are considered in order from best to worst. If given BS is included in partial solution and it is installed, then TP is assigned to it. If in partial solution it is not installed, the next BS from list is checked. If given BS is not included in partial solution, then TP is assigned to it. Then objective function is computed, including cost only for BS that partial solution uses, and the total power is calculated for assignment of BS described above.
4.4. Feasibility checking
Each partial solution is checked against feasibility. First it is checked (using assignment of TP from lower bound computing procedure) that every test point has been served. If this requirement is fulfilled, it could be expected that solutions in this branch will be able to serve all TPs. The next check that is performed, is number of TP assigned to each BS used by partial solution. If some BS has more than $max_{TP}$ assigned, it is considered as invalid solution. Then for each of the used BS the SIR is calculated, and if it is lower than $SIR_{min}$ then the partial solution is considered as invalid too.

4.5. Greedy score
For computing greedy score, the TP that can be served using BS included in partial solution need to be determined. Total served traffic $T_{max}$ is calculated, as the sum of simultaneous connection demand $u_i$ of TP that can be served. Then greedy score is simply calculated by (7):

$$greedyScore = lowerBound - T_{max}$$

(7)

5. Experimentation environment
As part of this work a new experimentation environment was developed following ideas proposed in [8]. The implemented simulator works with 2.5D and 3D vector data that can be imported into application from two different formats:

- Text format (used by Munich Test Site Building Data,
- CityGML (using libcitygml library).

When building data is imported, other information such as test points, BS candidate sites, frequency used etc. can be added and whole data saved into simulator's own binary file format.

Simulator user interface consists of four tabs:

- Scenario setup. Here user can setup change test points and BS candidate sites. New sites/test points can be added manually, or generated randomly. Main parameters of simulation can be changed here as well.
- Solver. Here user can calculate $G$ matrix and run experiments with implemented algorithms.
- Propagation prediction. Here user can compute signal strength prediction for regular grid of points. This can be useful to get general idea about range of specific BS on whole city area. The precision (number of grid points) can be set by user. The Signal strength information is visualised using colours.
- Full view. This tab is intended to explore scenario in 3D view.

In each of the first three tabs there is smaller scenario preview window. Every view of scenario can be manipulated using mouse and keyboard to change camera direction, zoom, and centre point. The experimentation environment allows changing the following parameters:

- Used frequency,
- Minimum signal-to-interference ratio (SIR) $SIR_{min}$,
- Maximum power of mobile equipment $P_{max}$,
- Minimum signal power requested at base station $P_{target}$,
- Ground level - useful for data which includes underground part,
- Criterion trade-off parameter $\lambda$.

Internally, binary space partitioning (BSP) tree is used to speed up geometric look-ups. For each transmitter and receiver position the faces intersecting on vertical plane between transmitter and receiver are found using BSP tree, then cross section between that plane and each of the faces is computed, and cross section of buildings is reconstructed. It is then used to find roofs, and estimate the
parameters of the empirical model. Then according to the equations given in section 6 the loss in dBm is computed. It can be then converted to normalised propagation gain using the following formula (8)

\[ g = \frac{1}{10^{10}} \]  

(8)

6. Empirical COST-Walfisch-Ikegami Model

There are many models created for prediction of radio wave propagation and signal strength. For the purposes of this work, the Empirical COST-Walfisch-Ikegami Model [9] was used. The motivation for choosing this model was simplicity and high prediction quality in urban environments. The considered model is a statistical model. The prediction is based on statistics of buildings in vertical plane between receiver and transmitter. The model captures propagation over the rooftops, which is main source of signal propagation in urban environments. The main parameters of the model are:

- Frequency \( f \) (800 . . . 2000 MHz),
- Height of the transmitter \( h_{TX} \) (4 . . . 50 m),
- Height of the receiver \( h_{RX}(1 . . . 3 \text{ m}) \),
- Distance \( d \) between transmitter and receiver (20 . . . 5000 m).

The following statistical parameters are estimated:

- Mean building height \( h_{\text{ROOF}} \),
- Mean width of street \( w \),
- Mean building separation \( b \).

The fourth parameter - the orientation of road \( \varphi \) - is omitted here because the orientation of the road cannot be easily determined for every point, especially on crossings.

The model distinguishes between two situations: the “line of sight” (LOS) and “no line of sight” (NLOS). In the LOS case, the propagation loss \( l_p \) between transmitter and receiver can be simply expressed by the formula (9):

\[ l_p = 42.6 + 26 \cdot \log \frac{d}{\text{km}} + 20 \cdot \log \frac{f}{\text{MHz}} \]  

(9)

In case of NLOS, the expression (10) is used:

\[ l_p = \begin{cases} l_0 + l_{\text{rts}} + l_{\text{msd}}, & l_{\text{rts}} + l_{\text{msd}} > 0 \\ l_0, & l_{\text{rts}} + l_{\text{msd}} \leq 0 \end{cases} \]  

(10)

where \( l_0 \) is free space loss, \( l_{\text{msd}} \) is multiple screen diffraction loss, and \( l_{\text{rts}} \) is the rooftop-to-street diffraction loss. The free space loss is calculated from formula (11):

\[ l_0 = 32.44 + 20 \cdot \log \frac{d}{\text{km}} + 20 \cdot \log \frac{f}{\text{MHz}} \]  

(11)

The rooftop-to-street diffraction loss term denoted by \( l_{\text{rts}} \) (12) describes the loss which occurs on the wave coupling into the street where the receiver is located.

\[ l_{\text{rts}} = -16.9 - 10 \cdot \log \frac{w}{m} + 10 \cdot \log \frac{f}{\text{MHz}} + 20 \cdot \log \frac{h_{\text{ROOF}} - h_{RX}}{m} + l_{\text{ori}} \]  

(12)
where $l_{diff}$ is omitted, because it depends on street orientation $\varphi$. It could be replaced by an empirical correction obtained from the calibration with measurements. An approximation for the multi-screen diffraction loss $l_{msd}$ is computed using the formula (13):

$$l_{msd} = l_{inh} + k_a + k_d \cdot \log \frac{d}{km} + k_f \cdot \log \frac{f}{MHz} - 9 \cdot \log \frac{p}{m}$$

with

$$l_{inh} = \begin{cases} 
-18 \cdot \left( 1 + \frac{h_{TX} - h_{ROOF}}{m} \right), & h_{TX} > h_{ROOF} \\
0, & h_{TX} \leq h_{ROOF} 
\end{cases}$$

$$k_a = \begin{cases} 
54, & h_{TX} > h_{ROOF} \\
54 - 0.8 \cdot \frac{h_{TX} - h_{ROOF}}{m}, & h_{TX} \leq h_{ROOF} / d \geq 0.5 \text{km} \\
54 - 0.8 \cdot \frac{h_{TX} - h_{ROOF} / d}{0.5}, & h_{TX} \leq h_{ROOF} / d < 0.5 \text{km}
\end{cases}$$

$$k_d = \begin{cases} 
18, & h_{TX} > h_{ROOF} \\
18 - 15 \cdot \frac{h_{TX} - h_{ROOF}}{h_{ROOF} - h_{RX}}, & h_{TX} \leq h_{ROOF}
\end{cases}$$

$$k_f = -4 + \left( \frac{f}{MHz} \right) \left( \frac{925}{f/MHz} - 1 \right), \begin{cases} 
0.7, & \text{for medium sized city} \\
1.5, & \text{for metropolitan centers}
\end{cases}$$

7. Investigation

The tests were conducted on 5 small instances characterized by 22 candidate sites with $u_i = 1$ (following data used in [3]). Additional two instances were generated, using 30 BS, and 80 TS (instance 6), and using 60 BS, and 130 TS (instance 7) – both instances uniformly distributed in {1, 2, 3}. The used parameters concerning frequency, interference ratio, and power are presented in table 1.

The parameters of greedy procedures and Tabu Search were used as in [3].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>2000 [MHz]</td>
</tr>
<tr>
<td>SIR$_{min}$</td>
<td>0.031250</td>
</tr>
<tr>
<td>$P_{max}$</td>
<td>30 [dBm]</td>
</tr>
<tr>
<td>$P_{target}$</td>
<td>-100 [dBm]</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Using the designed and implemented experimentation system described in section 5 – the simulation experiments were carried out. The five algorithms were compared:

- RGA (Randomized Greedy Add)
- RGR (Randomized Greedy Remove)
- MTSA (Multistart Tabu Search using RGA)
- MTSR (Multistart Tabu Search using RGR)
- B&B (Branch & Bound Algorithm)
The obtained results are shown in table 2, where in columns are presented: ‘cost’, i.e., the value of cost function defined by (1), and ‘#BS’ - the number of base stations in the solution, and in table 3 where in column is presented ‘time’, i.e., time of finding the solution. Note that the results for RGA for instances 6 and 7 are removed because a solution found was not feasible.

### Table 2. Cost comparison between algorithms.

<table>
<thead>
<tr>
<th>Inst</th>
<th>RGA Cost</th>
<th>RGA #BS</th>
<th>RGR Cost</th>
<th>RGR #BS</th>
<th>MTSA Cost</th>
<th>MTSA #BS</th>
<th>MTSR Cost</th>
<th>MTSR #BS</th>
<th>B&amp;B Cost</th>
<th>B&amp;B #BS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.89</td>
<td>7</td>
<td>2.21</td>
<td>7</td>
<td>1.17</td>
<td>10</td>
<td>1.19</td>
<td>10</td>
<td>0.36</td>
<td>21</td>
</tr>
<tr>
<td>2</td>
<td>2.54</td>
<td>5</td>
<td>3.99</td>
<td>8</td>
<td>1.18</td>
<td>11</td>
<td>1.79</td>
<td>8</td>
<td>0.36</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>2.49</td>
<td>8</td>
<td>2.04</td>
<td>7</td>
<td>1.02</td>
<td>10</td>
<td>1.05</td>
<td>10</td>
<td>0.33</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>1.21</td>
<td>10</td>
<td>1.44</td>
<td>14</td>
<td>1.11</td>
<td>11</td>
<td>1.20</td>
<td>13</td>
<td>0.65</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>1.31</td>
<td>8</td>
<td>1.76</td>
<td>9</td>
<td>0.84</td>
<td>10</td>
<td>0.80</td>
<td>10</td>
<td>0.22</td>
<td>22</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>1.41</td>
<td>20</td>
<td>1.08</td>
<td>16</td>
<td>1.05</td>
<td>19</td>
<td>0.64</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>1.64</td>
<td>40</td>
<td>1.28</td>
<td>32</td>
<td>1.24</td>
<td>28</td>
<td>0.80</td>
<td>50</td>
</tr>
</tbody>
</table>

### Table 3. Time comparison between algorithms.

<table>
<thead>
<tr>
<th>Inst</th>
<th>RGA [ms]</th>
<th>RGR [ms]</th>
<th>MTSA [s]</th>
<th>MTSR [s]</th>
<th>B&amp;B [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>198</td>
<td>200</td>
<td>160</td>
<td>154</td>
<td>17</td>
</tr>
<tr>
<td>2</td>
<td>119</td>
<td>182</td>
<td>128</td>
<td>125</td>
<td>83</td>
</tr>
<tr>
<td>3</td>
<td>204</td>
<td>288</td>
<td>184</td>
<td>187</td>
<td>97</td>
</tr>
<tr>
<td>4</td>
<td>285</td>
<td>180</td>
<td>197</td>
<td>197</td>
<td>138</td>
</tr>
<tr>
<td>5</td>
<td>199</td>
<td>173</td>
<td>224</td>
<td>228</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>635</td>
<td>616</td>
<td>465</td>
<td>2.8</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>635</td>
<td>4.5 [h]</td>
<td>3.07 [h]</td>
<td>302 [s]</td>
</tr>
</tbody>
</table>

It may be observed that RGA was unable to find feasible solution for bigger instances. The remaining algorithms were always able to find feasible solution. For RGR the time needed to obtain solution was not growing significantly with size of the instance, but the solutions found were worse that those obtained by more time consuming algorithms. The both variants of Tabu Search algorithms, MTSA and MTSR, delivered better solutions that randomised greedy RGA and RGR, but the computational time was growing significantly with larger size of instance.

The Branch & Bound algorithm was the only one delivering the exact solution. The solutions found were usually one order of magnitude better in terms of the cost function. For small instances the computational time was the smallest, however, it grows significantly with instance size, but not as much as for Tabu Search. One could also observe that Branch & Bound and also RGR for bigger instances, was using significantly more base stations that Tabu Search variants. This can be attributed to the fact, that for the cost function with $\lambda = 1$, the main objective was to minimise power emitted and the installation cost was negligible.

### 8. Conclusion

The Randomized greedy add procedure turned out to serve good trade-off between time spent on finding solution and quality of the solution obtained. The Tabu Search variants were far more computationally expensive and still were obtaining solution order of magnitude worse than optimal ones. The proposed Branch & Bound algorithm could be considered as promising – the obtained results of simulation experiments showed remarkable advantages. However, rapid increase of
computational time with the size of problem instance indicates that further improvement and extensions are needed. We are planning an improvement of the algorithm by applying hybrid meta-heuristics techniques, following ideas presented in [10] and [11]. Moreover, the problem defined and considered in this paper is one of the simpler ones and future work could focus on more detailed base station location problem models.

**Acknowledgement**

This work was supported by the statutory funds of the Faculty of Electronics, Department of Systems and Computer Networks, Wroclaw University of Technology, Wroclaw, Poland.

**References**