0.1 The linear programming formulation of the $L(j_1, j_2, ..., j_s)-$problem

1. $\min \lambda_{j_1,j_2,...,j_s}(G)$
2. $c(v) - c(u) + M \cdot z_{uv} \geq j_{dist(u,v)} \qquad \forall\, u, v \in V(G)$
3. $c(u) - c(v) + M \cdot (1 - z_{uv}) \geq j_{dist(u,v)} \qquad \forall\, u, v \in V(G)$
4. $z_{uv} \in \{0, 1\} \qquad \forall\, u, v \in V(G)$
5. $c(v) \geq 0 \qquad \forall\, v \in V(G)$
6. $c(v) \leq \lambda_{j_1,j_2,...,j_s}(G) \qquad \forall\, v \in V(G)$
7. $c(v) \in \mathbb{Z} \qquad \forall\, v \in V(G)$

0.2 Explication

In the formulation we should not use absolute value. We do not know at the time of the formulation which one of the values $c(u)$ or $c(v)$ will be greater. The binary variables z_{uv} are introduced to eliminate this problem: If $z_{uv} = 0$ then (2.) is valid if and only if $c(v) \geq c(u)$ and the restriction for them is satisfied, (3.) is satisfied because of the choice of M. If $z_{uv} = 1$ then (2.) is automatically valid and (3.) is valid if and only if $c(u) \geq c(v)$ and the restriction is satisfied.

So, the inequalities (2.)-(4.) are true together if and only if the restriction for the labels of v and u is met.

(5.) and (6.) describe that every node have to get a label between 0 and $\lambda_{j_1,j_2,...,j_s}(G)$.

(1.) is the objective function, it describes the main task.

The numbers $j_{dist(u,v)}$ are constants. If the distance between u and v is at most s then this constant appears in the labeling restriction. If the distance is bigger than s then $j_{dist(u,v)} = 0$.

The choice of M: One must choose a value for M which satisfies the inequalities (2.)-(3.) if the variables $c(v)$ take values such that they give a proper $L(j_1, j_2, ..., j_s)-$labeling of G. For that:

- $c(v) + M \cdot z_{uv} \geq c(u)$ and $c(u) + M \cdot (1 - z_{uv}) \geq c(v) \qquad \forall\, u, v \in V(G) \Longrightarrow M \geq |c(v) - c(u)| \qquad \forall\, u, v \in V(G)$
- $c(v) - c(u) + M \cdot z_{uv} \geq j_{dist(u,v)}$ and $c(u) - c(v) + M \cdot (1 - z_{uv}) \geq j_{dist(u,v)} \qquad \forall\, u, v \in V(G)$

$\Longrightarrow M \geq j_{dist(u,v)} + |c(u) - c(v)| \qquad \forall\, u, v \in V(G) \Longrightarrow M \geq j_1 + |c(u) - c(v)| \qquad \forall\, u, v \in V(G)$

The second restriction for the value of M is stricter hence $M \geq j_1 + \lambda_{j_1,j_2,...,j_s}(G)$ is right. But because $\lambda_{j_1,j_2,...,j_s}(G)$ is a variable in our formulation it is not allowed to be included in the declaration of M since M is the coefficient of some variables z_{uv}. To eliminate this problem write a trivial upper bound of $\lambda_{j_1,j_2,...,j_s}(G)$ instead of it. $(n - 1) \cdot j_1$ is clearly not bigger than the optimal labeling number. (In the case of a complete graph it is optimal). So $M = n \cdot j_1$ is a good choice.

The above formalization is proper in the sence that it describes the problem accurately. However, the efficiency is not good enough. There are a few possibilities to make the linear program better (Better in the sence that the computations become faster in average).

1 New models

The graph theoretical problem is a model for real radio systems where interference is avoided. In the classical model vertices correspond to transmitter locations and their labels (non-negative integers) to radio channels. Theoretically the graph is constructed in the way that the closer the vertices are in the graph, the closer the corresponding transmitter locations are in real. Of course, this model is just an approximation for the original problem, it does not portray that accurately. One of its major drawbacks is that the pair of vertices of same distance are considered as pair of transmitters of same distance. So it is possible that some pairs have an unnecessarily strict constraint, assumed the sufficient strictness for the whole graph due to the basic task. (Avoiding interference is more important than a small difference in the span.) In this model the more accurate representation can be achieved in two ways: Either introducing fictitious vertices (which increases the order and size of the graph), or increasing the number of constraints. Both procedures complicate the task significantly.

1.1 Possibilities for better models

- The graph and the discrete constraints remain unchanged but real values are allowed as labels. Since proper integer solutions solve this problem too, the minimum span is not bigger than that in the classical model.
- The graph remains unchanged but the edges are weighted, also real values are allowed as weights, as these describe the distance of the transmitters corresponding to the endvertices. In this case the distance of two non-adjacent vertices is still not accurate. However, the constraint can not be discrete anymore, so it should be given with a function on real domain.
- The graph is complete and for each pair of vertices their Euclidean distance is considered. The constraints are given in a real valued function on real domains, the labels may have real values as well. This model is the most precize reprezentation of the original problem. (I call this version the Euclidean model.)

In the case of finite graphs there are finite and discrete number of pairs of vertices, so the constraints can be given in the classical way.

1.2 Testing the models

Formulation

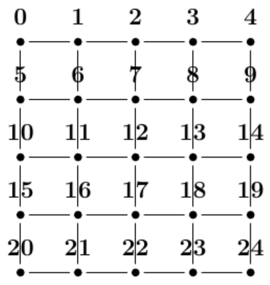
The formulation should differ from the one for the classical model only in two little points, namely a function $\ell(dist(u,v))$ should be used instead of $j_{dist(u,v)}$, and the c values are allowed to be real numbers. While $j_{dist(u,v)}$ is a function with $\mathbb{N} \mapsto \mathbb{N}$, according to the classical model, $\ell(dist(u,v))$ is a function with $\mathbb{R} \mapsto \mathbb{R}$. So, our formulation for the problem in the Eucledean model is the following:

1. $\min \lambda_\ell(G)$
2. $c(v) - c(u) + M \cdot z_{uv} \geq \ell(dist(u,v)) \qquad \forall\, u, v \in V(G)$
3. $c(u) - c(v) + M \cdot (1 - z_{uv}) \geq \ell(dist(u,v)) \qquad \forall\, u, v \in V(G)$
4. $z_{uv} \in \{0,1\} \qquad \forall\, u, v \in V(G)$
5. $c(v) \geq 0 \qquad \forall\, v \in V(G)$
6. $c(v) \leq \lambda_\ell(G) \qquad \forall\, v \in V(G)$
7. $c(v) \in \mathbb{R} \qquad \forall\, v \in V(G)$

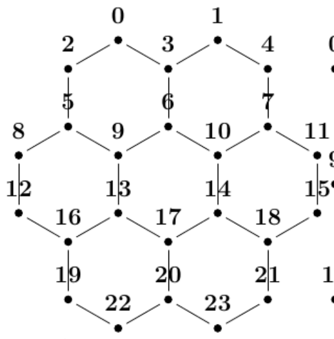
Lattice graphs

In practice three lattices have prominent role: the hexagonal, the triangular and the square lattices. In the radio and mobile networks the large areas are covered by these polygons. Experience has shown that the covering by hexagonals is the most economical one. In this case the transmitters are in the centers of the hexagonals, that are considered as the vertices of the lattice graph, and are adjacent if and only if the corresponding hexagonals share a common edge. The graph constructed this way is a triangular lattice.

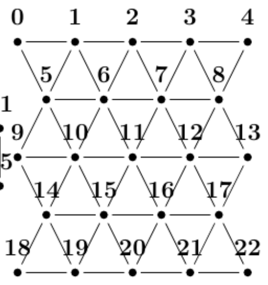
The lattices have the property that for a given size there are only a few distances occuring in them. This is the case in all of the above models. In the following one can see some distance matrices of the lattices of size 24, 23 and 25. The first matrices contain the euklidean distances, and the second matrices contain the graph distances. The values of the minimum spans have been searched by the software CPLEX. Since the graph distances are in a lot of cases higher than the Euclidean distances, the same general constraints are stricter for the case when Euclidean distances are considered. That is the reason why the spans are higher then. After modificating the constraints so that these are the same for every pair of vertices of course the results are the same. In some cases the new values are even lower. These modifications can be done because of the relatively small number of distances occuring.



a) Square Lattice



b) Hexagonal Lattice



c) Triangular Lattice

1.2.1 Hexagonal latice

The Euclidean distances

0	$\sqrt{3}$	1	1	$\sqrt{7}$	$\sqrt{3}$	$\sqrt{3}$	3	$\sqrt{7}$	2	$\sqrt{7}$	4	$2\sqrt{3}$	3	$2\sqrt{3}$	$\sqrt{21}$	$\sqrt{13}$	$\sqrt{13}$	$\sqrt{19}$	$\sqrt{21}$	$\sqrt{21}$	$\sqrt{27}$	5	$2\sqrt{7}$
$\sqrt{3}$	0	$\sqrt{7}$	1	1	3	$\sqrt{3}$	$\sqrt{3}$	4	$\sqrt{7}$	2	$\sqrt{7}$	$\sqrt{21}$	$2\sqrt{3}$	3	$2\sqrt{3}$	$\sqrt{19}$	$\sqrt{13}$	$\sqrt{13}$	$\sqrt{27}$	$\sqrt{21}$	$\sqrt{21}$	$2\sqrt{7}$	5
1	$\sqrt{7}$	0	$\sqrt{3}$	$2\sqrt{3}$	1	2	$\sqrt{13}$	$\sqrt{3}$	$\sqrt{3}$	3	$\sqrt{21}$	$\sqrt{7}$	$\sqrt{7}$	$\sqrt{13}$	5	3	$2\sqrt{3}$	$\sqrt{21}$	4	$\sqrt{19}$	$2\sqrt{7}$	$\sqrt{21}$	$\sqrt{27}$
1	1	$\sqrt{3}$	0	$\sqrt{3}$	2	1	2	3	$\sqrt{3}$	$\sqrt{3}$	3	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{7}$	$\sqrt{13}$	$2\sqrt{3}$	3	$2\sqrt{3}$	$\sqrt{19}$	4	$\sqrt{19}$	$\sqrt{21}$	$\sqrt{21}$
$\sqrt{7}$	1	$2\sqrt{3}$	$\sqrt{3}$	0	$\sqrt{13}$	2	1	$\sqrt{21}$	3	$\sqrt{3}$	$\sqrt{3}$	5	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{7}$	$\sqrt{21}$	$2\sqrt{3}$	3	$2\sqrt{7}$	$\sqrt{19}$	4	$\sqrt{27}$	$\sqrt{21}$
$\sqrt{3}$	3	1	2	$\sqrt{13}$	0	3	$2\sqrt{3}$	1	1	$\sqrt{7}$	$\sqrt{19}$	$\sqrt{3}$	$\sqrt{3}$	$\sqrt{13}$	$\sqrt{21}$	2	$\sqrt{7}$	4	3	$2\sqrt{3}$	$\sqrt{21}$	$\sqrt{13}$	$\sqrt{19}$
$\sqrt{3}$	$\sqrt{3}$	2	1	2	$\sqrt{3}$	0	$\sqrt{3}$	$\sqrt{7}$	1	1	$\sqrt{7}$	3	$\sqrt{3}$	$\sqrt{3}$	3	$\sqrt{7}$	2	$\sqrt{7}$	$2\sqrt{3}$	3	$2\sqrt{3}$	$\sqrt{13}$	$\sqrt{13}$
3	$\sqrt{3}$	$\sqrt{13}$	2	1	$2\sqrt{3}$	$\sqrt{3}$	0	$\sqrt{19}$	$\sqrt{7}$	1	1	$\sqrt{21}$	3	$\sqrt{3}$	$\sqrt{3}$	4	$\sqrt{7}$	2	$\sqrt{21}$	$2\sqrt{3}$	3	$\sqrt{19}$	$\sqrt{13}$
$\sqrt{7}$	4	$\sqrt{3}$	3	$\sqrt{21}$	1	$\sqrt{7}$	$\sqrt{19}$	0	$\sqrt{3}$	$2\sqrt{3}$	$3\sqrt{3}$	1	2	$\sqrt{13}$	$2\sqrt{7}$	$\sqrt{3}$	3	$\sqrt{21}$	$\sqrt{7}$	$\sqrt{13}$	5	$2\sqrt{3}$	$\sqrt{21}$
2	$\sqrt{7}$	$\sqrt{3}$	$\sqrt{3}$	3	1	1	$\sqrt{7}$	$\sqrt{3}$	0	$\sqrt{3}$	$2\sqrt{3}$	2	1	2	$\sqrt{13}$	$\sqrt{3}$	$\sqrt{3}$	3	$\sqrt{7}$	$\sqrt{7}$	$\sqrt{13}$	3	$2\sqrt{3}$
$\sqrt{7}$	2	3	$\sqrt{3}$	$\sqrt{3}$	$\sqrt{7}$	1	1	$2\sqrt{3}$	$\sqrt{3}$	0	$\sqrt{3}$	$\sqrt{13}$	2	1	2	3	$\sqrt{3}$	$\sqrt{3}$	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{7}$	$2\sqrt{3}$	3
4	$\sqrt{7}$	$\sqrt{21}$	3	$\sqrt{3}$	$\sqrt{19}$	$\sqrt{7}$	1	$3\sqrt{3}$	$2\sqrt{3}$	$\sqrt{3}$	0	$2\sqrt{7}$	$\sqrt{13}$	2	1	$\sqrt{21}$	3	$\sqrt{3}$	5	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{21}$	$2\sqrt{3}$
$2\sqrt{3}$	$\sqrt{21}$	$\sqrt{7}$	$\sqrt{13}$	5	$\sqrt{3}$	3	$\sqrt{21}$	1	2	$\sqrt{13}$	$2\sqrt{7}$	0	$\sqrt{3}$	$2\sqrt{3}$	$3\sqrt{3}$	1	$\sqrt{7}$	$\sqrt{19}$	$\sqrt{3}$	3	$\sqrt{21}$	$\sqrt{7}$	4
3	$2\sqrt{3}$	$\sqrt{7}$	$\sqrt{7}$	$\sqrt{13}$	$\sqrt{3}$	$\sqrt{3}$	3	2	1	2	$\sqrt{13}$	$\sqrt{3}$	0	$\sqrt{3}$	$2\sqrt{3}$	1	1	$\sqrt{7}$	$\sqrt{3}$	$\sqrt{3}$	3	2	$\sqrt{7}$
$2\sqrt{3}$	3	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{7}$	$\sqrt{13}$	$\sqrt{3}$	$\sqrt{3}$	$\sqrt{13}$	2	1	2	$2\sqrt{3}$	$\sqrt{3}$	0	$\sqrt{3}$	$\sqrt{7}$	1	1	3	$\sqrt{3}$	$\sqrt{3}$	$\sqrt{7}$	2
$\sqrt{21}$	$2\sqrt{3}$	5	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{21}$	3	$\sqrt{3}$	$2\sqrt{7}$	$\sqrt{13}$	2	1	$3\sqrt{3}$	$2\sqrt{3}$	$\sqrt{3}$	0	$\sqrt{19}$	$\sqrt{7}$	1	$\sqrt{21}$	3	$\sqrt{3}$	4	$\sqrt{7}$
$\sqrt{13}$	$\sqrt{19}$	3	$2\sqrt{3}$	$\sqrt{21}$	2	$\sqrt{7}$	4	$\sqrt{3}$	$\sqrt{3}$	3	$\sqrt{21}$	1	1	$\sqrt{7}$	$\sqrt{19}$	0	$\sqrt{3}$	$2\sqrt{3}$	1	2	$\sqrt{13}$	$\sqrt{3}$	3
$\sqrt{13}$	$\sqrt{13}$	$2\sqrt{3}$	3	$2\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{7}$	3	$\sqrt{3}$	$\sqrt{3}$	3	$\sqrt{7}$	1	1	$\sqrt{7}$	$\sqrt{3}$	0	$\sqrt{3}$	2	1	2	$\sqrt{3}$	$\sqrt{3}$
$\sqrt{19}$	$\sqrt{13}$	$\sqrt{21}$	$2\sqrt{3}$	3	4	$\sqrt{7}$	2	$\sqrt{21}$	3	$\sqrt{3}$	$\sqrt{3}$	$\sqrt{19}$	$\sqrt{7}$	1	1	$2\sqrt{3}$	$\sqrt{3}$	0	$\sqrt{13}$	2	1	3	$\sqrt{3}$
$\sqrt{21}$	$\sqrt{27}$	4	$\sqrt{19}$	$2\sqrt{7}$	3	$2\sqrt{3}$	$\sqrt{21}$	$\sqrt{7}$	$\sqrt{7}$	$\sqrt{13}$	5	$\sqrt{3}$	$\sqrt{3}$	3	$\sqrt{21}$	1	2	$\sqrt{13}$	0	$\sqrt{3}$	$2\sqrt{3}$	1	$\sqrt{7}$
$\sqrt{21}$	$\sqrt{21}$	$\sqrt{19}$	4	$\sqrt{19}$	$2\sqrt{3}$	3	$2\sqrt{3}$	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{7}$	$\sqrt{13}$	3	$\sqrt{3}$	$\sqrt{3}$	3	2	1	2	$\sqrt{3}$	0	$\sqrt{3}$	1	1
$\sqrt{27}$	$\sqrt{21}$	$2\sqrt{7}$	$\sqrt{19}$	4	$\sqrt{21}$	$2\sqrt{3}$	3	5	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{7}$	$\sqrt{21}$	$\sqrt{7}$	$\sqrt{3}$	$\sqrt{3}$	$\sqrt{13}$	2	1	$2\sqrt{3}$	$\sqrt{3}$	0	$\sqrt{7}$	1
5	$2\sqrt{7}$	$\sqrt{21}$	$\sqrt{21}$	$\sqrt{27}$	$\sqrt{13}$	$\sqrt{13}$	$\sqrt{19}$	$2\sqrt{3}$	3	$2\sqrt{3}$	$\sqrt{21}$	$\sqrt{7}$	2	$\sqrt{7}$	4	$\sqrt{3}$	$\sqrt{3}$	3	1	1	$\sqrt{7}$	0	$\sqrt{3}$
$2\sqrt{7}$	5	$\sqrt{27}$	$\sqrt{21}$	$\sqrt{21}$	$\sqrt{19}$	$\sqrt{13}$	$\sqrt{13}$	$\sqrt{21}$	$2\sqrt{3}$	3	$2\sqrt{3}$	4	$\sqrt{7}$	2	$\sqrt{7}$	3	$\sqrt{3}$	$\sqrt{3}$	$\sqrt{7}$	1	1	$\sqrt{3}$	0

The graph distances

0	2	1	1	3	2	2	4	3	3	3	5	4	4	4	6	5	5	5	6	6	6	7	7
2	0	3	1	1	4	2	2	5	3	3	3	6	4	4	4	5	5	5	6	6	6	7	7
1	3	0	2	4	1	3	5	2	2	4	6	3	3	5	7	4	4	6	5	5	7	6	6
1	1	2	0	2	3	1	3	4	2	2	4	5	3	3	5	4	4	4	5	5	5	6	6
3	1	4	2	0	5	3	1	6	4	2	2	7	5	3	3	6	4	4	7	5	5	6	6
2	4	1	3	5	0	2	4	1	1	3	5	2	2	4	6	3	3	5	4	4	6	5	5
2	2	3	1	3	2	0	2	3	1	1	3	4	2	2	4	3	3	3	4	4	4	5	5
4	2	5	3	1	4	2	0	5	3	1	1	6	4	2	2	5	3	3	6	4	4	5	5
3	5	2	4	6	1	3	5	0	2	4	6	1	3	5	7	2	4	6	3	5	7	4	6
3	3	2	2	4	1	1	3	2	0	2	4	3	1	3	5	2	2	4	3	3	5	4	4
3	3	4	2	2	3	1	1	4	2	0	2	5	3	1	3	4	2	2	5	3	3	4	4
5	3	6	4	2	5	3	1	6	4	2	0	7	5	3	1	6	4	2	7	5	3	6	4
4	6	3	5	7	2	4	6	1	3	5	7	0	2	4	6	1	3	5	2	4	6	3	5
4	4	3	3	5	2	2	4	3	1	3	5	2	0	2	4	1	1	3	2	2	4	3	3
4	4	5	3	3	4	2	2	5	3	1	3	4	2	0	2	3	1	1	4	2	2	3	3
6	4	7	5	3	6	4	2	7	5	3	1	6	4	2	0	5	3	1	6	4	2	5	3
5	5	4	4	6	3	3	5	2	2	4	6	1	1	3	5	0	2	4	1	3	5	2	4
5	5	4	4	4	3	3	3	4	2	2	4	3	1	1	3	2	0	2	3	1	3	2	2
5	5	6	4	4	5	3	3	6	4	2	2	5	3	1	1	4	2	0	5	3	1	4	2
6	6	5	5	7	4	4	6	3	3	5	7	2	2	4	6	1	3	5	0	2	4	1	3
6	6	5	5	5	4	4	4	5	3	3	5	4	2	2	4	3	1	3	2	0	2	1	1
6	6	7	5	5	6	4	4	7	5	3	3	6	4	2	2	5	3	1	4	2	0	3	1
7	7	6	6	6	5	5	5	4	4	4	6	3	3	3	5	2	2	4	1	1	3	0	2
7	7	6	6	6	5	5	5	6	4	4	4	5	3	3	3	4	2	2	3	1	1	2	0

1.2.2 The triangular lattice

The Euclidean distances

0	1	2	3	4	1	$\sqrt{3}$	$\sqrt{7}$	$\sqrt{13}$	$\sqrt{3}$	2	$\sqrt{7}$	$2\sqrt{3}$	$\sqrt{19}$	$\sqrt{7}$	3	$\sqrt{13}$	$\sqrt{19}$	$2\sqrt{3}$	$\sqrt{13}$	4	$\sqrt{21}$	$2\sqrt{7}$
1	0	1	2	3	1	1	$\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	2	$\sqrt{7}$	$2\sqrt{3}$	$\sqrt{7}$	$\sqrt{7}$	3	$\sqrt{13}$	$\sqrt{13}$	$2\sqrt{3}$	$\sqrt{13}$	4	$\sqrt{21}$
2	1	0	1	2	$\sqrt{3}$	1	1	$\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	2	$\sqrt{7}$	3	$\sqrt{7}$	$\sqrt{7}$	3	4	$\sqrt{13}$	$2\sqrt{3}$	$\sqrt{13}$	4
3	2	1	0	1	$\sqrt{7}$	$\sqrt{3}$	1	1	$2\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	2	$\sqrt{13}$	3	$\sqrt{7}$	$\sqrt{7}$	$\sqrt{21}$	4	$\sqrt{13}$	$2\sqrt{3}$	$\sqrt{13}$
4	3	2	1	0	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{3}$	1	$\sqrt{19}$	$2\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	$\sqrt{19}$	$\sqrt{13}$	3	$\sqrt{7}$	$2\sqrt{7}$	$\sqrt{21}$	4	$\sqrt{13}$	$2\sqrt{3}$
1	1	$\sqrt{3}$	$\sqrt{7}$	$\sqrt{13}$	0	1	2	3	1	1	$\sqrt{3}$	$\sqrt{7}$	$\sqrt{13}$	$\sqrt{3}$	2	$\sqrt{7}$	$2\sqrt{3}$	$\sqrt{7}$	$\sqrt{7}$	3	$\sqrt{13}$	$\sqrt{19}$
$\sqrt{3}$	1	1	$\sqrt{3}$	$\sqrt{7}$	1	0	1	2	$\sqrt{3}$	1	1	$\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	2	$\sqrt{7}$	3	$\sqrt{13}$	$\sqrt{13}$	3	$\sqrt{13}$
$\sqrt{7}$	$\sqrt{3}$	1	1	$\sqrt{3}$	2	1	0	1	$\sqrt{7}$	$\sqrt{3}$	1	1	$\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	2	$\sqrt{13}$	3	$\sqrt{7}$	$\sqrt{7}$	3
$\sqrt{13}$	$\sqrt{7}$	$\sqrt{3}$	1	1	3	2	1	0	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{3}$	1	1	$2\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	$\sqrt{19}$	$\sqrt{13}$	3	$\sqrt{7}$	$\sqrt{7}$
$\sqrt{3}$	2	$\sqrt{7}$	$2\sqrt{3}$	$\sqrt{19}$	1	$\sqrt{3}$	$\sqrt{7}$	$\sqrt{13}$	0	1	2	3	4	1	$\sqrt{3}$	$\sqrt{7}$	$\sqrt{13}$	$\sqrt{3}$	2	$\sqrt{7}$	$2\sqrt{3}$	$\sqrt{19}$
2	$\sqrt{3}$	2	$\sqrt{7}$	$2\sqrt{3}$	1	1	$\sqrt{3}$	$\sqrt{7}$	1	0	1	2	3	1	1	$\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	2	$\sqrt{7}$	$2\sqrt{3}$
$\sqrt{7}$	2	$\sqrt{3}$	2	$\sqrt{7}$	$\sqrt{3}$	1	1	$\sqrt{3}$	2	1	0	1	2	$\sqrt{3}$	1	1	$\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	2	$\sqrt{7}$
$2\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	2	$\sqrt{7}$	$\sqrt{3}$	1	1	3	2	1	0	1	$\sqrt{7}$	$\sqrt{3}$	1	1	$2\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	2
$\sqrt{19}$	$2\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{3}$	1	4	3	2	1	0	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{3}$	1	$\sqrt{19}$	$2\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$
$\sqrt{7}$	$\sqrt{7}$	3	$\sqrt{13}$	$\sqrt{19}$	$\sqrt{3}$	2	$\sqrt{7}$	$2\sqrt{3}$	1	1	$\sqrt{3}$	$\sqrt{7}$	$\sqrt{13}$	0	1	2	3	1	1	$\sqrt{3}$	$\sqrt{7}$	$\sqrt{13}$
3	$\sqrt{7}$	$\sqrt{7}$	3	$\sqrt{13}$	2	$\sqrt{3}$	2	$\sqrt{7}$	$\sqrt{3}$	1	1	$\sqrt{3}$	$\sqrt{7}$	1	0	1	2	$\sqrt{3}$	1	1	$\sqrt{3}$	$\sqrt{7}$
$\sqrt{13}$	3	$\sqrt{7}$	$\sqrt{7}$	3	$\sqrt{7}$	2	$\sqrt{3}$	2	$\sqrt{7}$	$\sqrt{3}$	1	1	$\sqrt{3}$	2	1	0	1	$\sqrt{7}$	$\sqrt{3}$	1	1	$\sqrt{3}$
$\sqrt{19}$	$\sqrt{13}$	3	$\sqrt{7}$	$\sqrt{7}$	$2\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{3}$	1	1	3	2	1	0	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{3}$	1	1
$2\sqrt{3}$	$\sqrt{13}$	4	$\sqrt{21}$	$2\sqrt{7}$	$\sqrt{7}$	3	$\sqrt{13}$	$\sqrt{19}$	$\sqrt{3}$	2	$\sqrt{7}$	$2\sqrt{3}$	$\sqrt{19}$	1	$\sqrt{3}$	$\sqrt{7}$	$\sqrt{13}$	0	1	2	3	4
$\sqrt{13}$	$2\sqrt{3}$	$\sqrt{13}$	4	$\sqrt{21}$	$\sqrt{7}$	$\sqrt{13}$	3	$\sqrt{13}$	2	$\sqrt{3}$	2	$\sqrt{7}$	$2\sqrt{3}$	1	1	$\sqrt{3}$	$\sqrt{7}$	1	0	1	2	3
4	$\sqrt{13}$	$2\sqrt{3}$	$\sqrt{13}$	4	3	$\sqrt{13}$	$\sqrt{7}$	3	$\sqrt{7}$	2	$\sqrt{3}$	2	$\sqrt{7}$	$\sqrt{3}$	1	1	$\sqrt{3}$	2	1	0	1	2
$\sqrt{21}$	4	$\sqrt{13}$	$2\sqrt{3}$	$\sqrt{13}$	$\sqrt{13}$	3	$\sqrt{7}$	$\sqrt{7}$	$2\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	2	$\sqrt{7}$	$\sqrt{3}$	1	1	3	2	1	0	1
$2\sqrt{7}$	$\sqrt{21}$	4	$\sqrt{13}$	$2\sqrt{3}$	$\sqrt{19}$	$\sqrt{13}$	3	$\sqrt{7}$	$\sqrt{19}$	$2\sqrt{3}$	$\sqrt{7}$	2	$\sqrt{3}$	$\sqrt{13}$	$\sqrt{7}$	$\sqrt{3}$	1	4	3	2	1	0

The graph distances

0	1	2	3	4	1	2	3	4	2	2	3	4	5	3	3	4	5	4	4	4	5	6
1	0	1	2	3	1	1	2	3	2	2	2	3	4	3	3	3	4	4	4	4	4	5
2	1	0	1	2	2	1	1	2	3	2	2	2	3	3	3	3	3	4	4	4	4	4
3	2	1	0	1	3	2	1	1	4	3	2	2	2	4	3	3	3	5	4	4	4	4
4	3	2	1	0	4	3	2	1	5	4	3	2	2	5	4	3	3	6	5	4	4	4
1	1	2	3	4	0	1	2	3	1	1	2	3	4	2	2	3	4	3	3	3	4	5
2	1	1	2	3	1	0	1	2	2	1	1	2	3	2	2	2	3	3	3	3	3	4
3	2	1	1	2	2	1	0	1	3	2	1	1	2	3	2	2	2	4	3	3	3	3
4	3	2	1	1	3	2	1	0	4	3	2	1	1	4	3	2	2	5	4	3	3	3
2	2	3	4	5	1	2	3	4	0	1	2	3	4	1	2	3	4	2	2	3	4	5
2	2	2	3	4	1	1	2	3	1	0	1	2	3	1	1	2	3	2	2	2	3	4
3	2	2	2	3	2	1	1	2	2	1	0	1	2	2	1	1	2	3	2	2	2	3
4	3	2	2	2	3	2	1	1	3	2	1	0	1	3	2	1	1	4	3	2	2	2
5	4	3	2	2	4	3	2	1	4	3	2	1	0	4	3	2	1	5	4	3	2	2
3	3	3	4	5	2	2	3	4	1	1	2	3	4	0	1	2	3	1	1	2	3	4
3	3	3	3	4	2	2	2	3	2	1	1	2	3	1	0	1	2	2	1	1	2	3
4	3	3	3	3	3	2	2	2	3	2	1	1	2	2	1	0	1	3	2	1	1	2
5	4	3	3	3	4	3	2	2	4	3	2	1	1	3	2	1	0	4	3	2	1	1
4	4	4	5	6	3	3	4	5	2	2	3	4	5	1	2	3	4	0	1	2	3	4
4	4	4	4	5	3	3	3	4	2	2	2	3	4	1	1	2	3	1	0	1	2	3
4	4	4	4	4	3	3	3	3	3	2	2	2	3	2	1	1	2	2	1	0	1	2
5	4	4	4	4	4	3	3	3	4	3	2	2	2	3	2	1	1	3	2	1	0	1
6	5	4	4	4	5	4	3	3	5	4	3	2	2	4	3	2	1	4	3	2	1	0

1.2.3 The square lattice

The Euclidean distances

0	1	2	3	4	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{17}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{20}$	3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{18}$	5	4	$\sqrt{17}$	$\sqrt{20}$	5	$\sqrt{32}$
1	0	1	2	3	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{18}$	$\sqrt{17}$	4	$\sqrt{17}$	$\sqrt{20}$	5
2	1	0	1	2	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{20}$	$\sqrt{17}$	4	$\sqrt{17}$	$\sqrt{20}$
3	2	1	0	1	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{18}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$	5	$\sqrt{20}$	$\sqrt{17}$	4	$\sqrt{17}$
4	3	2	1	0	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{20}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	5	$\sqrt{18}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{32}$	5	$\sqrt{20}$	$\sqrt{17}$	4
1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{17}$	0	1	2	3	4	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{17}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{20}$	3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{18}$	5
$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	1	0	1	2	3	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{18}$
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	2	1	0	1	2	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$
$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	3	2	1	0	1	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{18}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$
$\sqrt{17}$	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	4	3	2	1	0	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{20}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	5	$\sqrt{18}$	$\sqrt{13}$	$\sqrt{10}$	3
2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{20}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{17}$	0	1	2	3	4	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{17}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{20}$
$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	1	0	1	2	3	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$
$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	2	1	0	1	2	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$
$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	3	2	1	0	1	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$
$\sqrt{20}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	4	3	2	1	0	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{20}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2
3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{18}$	5	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{20}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{17}$	0	1	2	3	4	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{17}$
$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{18}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	1	0	1	2	3	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$
$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	2	1	0	1	2	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
$\sqrt{18}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	3	2	1	0	1	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$
5	$\sqrt{18}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{20}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	4	3	2	1	0	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1
4	$\sqrt{17}$	$\sqrt{20}$	5	$\sqrt{32}$	3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{18}$	5	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{20}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{17}$	0	1	2	3	4
$\sqrt{17}$	4	$\sqrt{17}$	$\sqrt{20}$	5	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{18}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{13}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	$\sqrt{10}$	1	0	1	2	3
$\sqrt{20}$	$\sqrt{17}$	4	$\sqrt{17}$	$\sqrt{20}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$	2	1	0	1	2
5	$\sqrt{20}$	$\sqrt{17}$	4	$\sqrt{17}$	$\sqrt{18}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{10}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	3	2	1	0	1
$\sqrt{32}$	5	$\sqrt{20}$	$\sqrt{17}$	4	5	$\sqrt{18}$	$\sqrt{13}$	$\sqrt{10}$	3	$\sqrt{20}$	$\sqrt{13}$	$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	1	4	3	2	1	0

The graph distances

0	1	2	3	4	1	2	3	4	5	2	3	4	5	6	3	4	5	6	7	4	5	6	7	8
1	0	1	2	3	2	1	2	3	4	3	2	3	4	5	4	3	4	5	6	5	4	5	6	7
2	1	0	1	2	3	2	1	2	3	4	3	2	3	4	5	4	3	4	5	6	5	4	5	6
3	2	1	0	1	4	3	2	1	2	5	4	3	2	3	6	5	4	3	4	7	6	5	4	5
4	3	2	1	0	5	4	3	2	1	6	5	4	3	2	7	6	5	4	3	8	7	6	5	4
1	2	3	4	5	0	1	2	3	4	1	2	3	4	5	2	3	4	5	6	3	4	5	6	7
2	1	2	3	4	1	0	1	2	3	2	1	2	3	4	3	2	3	4	5	4	3	4	5	6
3	2	1	2	3	2	1	0	1	2	3	2	1	2	3	4	3	2	3	4	5	4	3	4	5
4	3	2	1	2	3	2	1	0	1	4	3	2	1	2	5	4	3	2	3	6	5	4	3	4
5	4	3	2	1	4	3	2	1	0	5	4	3	2	1	6	5	4	3	2	7	6	5	4	3
2	3	4	5	6	1	2	3	4	5	0	1	2	3	4	1	2	3	4	5	2	3	4	5	6
3	2	3	4	5	2	1	2	3	4	1	0	1	2	3	2	1	2	3	4	3	2	3	4	5
4	3	2	3	4	3	2	1	2	3	2	1	0	1	2	3	2	1	2	3	4	3	2	3	4
5	4	3	2	3	4	3	2	1	2	3	2	1	0	1	4	3	2	1	2	5	4	3	2	3
6	5	4	3	2	5	4	3	2	1	4	3	2	1	0	5	4	3	2	1	6	5	4	3	2
3	4	5	6	7	2	3	4	5	6	1	2	3	4	5	0	1	2	3	4	1	2	3	4	5
4	3	4	5	6	3	2	3	4	5	2	1	2	3	4	1	0	1	2	3	2	1	2	3	4
5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	1	2	3	2	1	2	3
6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	1	4	3	2	1	2
7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0	5	4	3	2	1
4	5	6	7	8	3	4	5	6	7	2	3	4	5	6	1	2	3	4	5	0	1	2	3	4
5	4	5	6	7	4	3	4	5	6	3	2	3	4	5	2	1	2	3	4	1	0	1	2	3
6	5	4	5	6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	1	2
7	6	5	4	5	6	5	4	3	4	5	4	3	2	3	4	3	2	1	2	3	2	1	0	1
8	7	6	5	4	7	6	5	4	3	6	5	4	3	2	5	4	3	2	1	4	3	2	1	0

The path in the graph between two vertices is a broken line. Of course this is longer than the line segment joining the vertices (their Euclidean distance). The degree of difference varies, it can be relatively small or zero, but also very high. (For example $\sqrt{32} \leftrightarrow 8$, the multiplier is $\sqrt{2}$.) Because of the variability of degree the restrictive conditions may not be alleviated. It is not even appropriate to alleviate them at difference rates for the individual distances. (For example two vertices with graph distance 4 can have Euclidean distance 4, $\sqrt{10}$ and $\sqrt{8}$ as well.)

1.3 The computations that were done for each one of the 3 above lattices

We compared the classical and the Euclidean model in the aspects of labeling number and running time of the computations. The goal is to compare the calculations in the more precize model with these in the classic model. In the classical model every distance is an integer. The constraint should be reworded so that the same applies to alldistances. The function values in the non-integer points do not matter, the only important thing is that the function is monotonically decreasing. So for the case $L(j, j-1, ..., 1)-$labeling $\ell(dist) = j + 1 - dist$ is a good choice.

1.3.1 The linear functions

At first we chose the linear functions $\ell_2(dist) = 3 - dist$, $\ell_3(dist) = 4 - dist$. These correspond to the $L(2,1)-$, $L(3,2,1)-$labeling, respectively, in the classical model.

The function values in the points, where *dist* (the Euclidean distance) is a natural number, are the same as in the classical model. Nevertheless, the constraint is not the same, because the distances are in general not the same. The distance of adjacent vertices is 1 in both cases, but that of non-adjacent vertices can be the same as their graph distance, but also much smaller. Because of this the restrictions for the vertex pairs are at least as strict as in the classical model, but for the most pairs tightly stricter. So, it is not surprising, that the labeling numbers are bigger. In the following table we present the labeling numbers that were computed by CPLEX, and the running times that were needed for the computations.

Lattice	$\ell_2(dist) = 3 - dist$	$L(2,1)$	$\ell_3(dist) = 4 - dist$	$L(3,2,1)$
Hexagonal	6.41699 (1.69 s)	5 (0.08 s)	16.6243 (1838.73 s)	9 (0.29 s)
Triangular	9.78029 (10.96 s)	8 (0.34 s)	21.9068 (103883 s)	18 (167.29 s)
Square	8.63494 (4.92 s)	6 (0.2 s)	19.9067 (27793.8 s)	11 (0.68 s)

As the table shows, not just the labeling numbers are bigger in the Euclidean model, but also the running times. Although the linear functions are rather equivalent to the discrete constraints, we need some additional ones for the appropriate comparisons.

1.3.2 Labeling with step functions

Let be the constraint defining functions such step functions, which take values so, that the constraints for the most of the pairs of vertices are the same as in the classical model. (Just for most of the pairs and not for all of them). The step functions have been constructed in the following way: Supposed that the constraints of the classical model are chosen correctly, one changes the constraints of the Euclidean model, to achieve about the same severity. In the following tables the occurring Euclidean distances are shown. In addition, the graph distances that the corresponding vertex pairs have in the classical model.

Hexagonal:

1	1
$\sqrt{3}$	2
$2, \sqrt{7}$	3
$3, 2\sqrt{3}$	4
$4, \sqrt{13}, \sqrt{19}$	5
$\sqrt{21}, 3\sqrt{3}$	6
$5, 2\sqrt{7}$	7

$$\ell(dist) = \begin{cases} 7 & 0 < dist \leq 1 \\ 6 & 1 < dist \leq \sqrt{3} \\ 5 & \sqrt{3} < dist \leq \sqrt{7} \\ 4 & \sqrt{7} < dist \leq 2\sqrt{3} \\ 3 & \text{if } 2\sqrt{3} < dist \leq \sqrt{19} \\ 2 & \sqrt{19} < dist \leq 3\sqrt{3} \\ 1 & 3\sqrt{3} < dist \leq 2\sqrt{7} \\ 0 & 2\sqrt{7} < dist \end{cases}$$

Exceptions: There are some vertex pairs with Euclidean distance $3\sqrt{3}$ and graph distance 6, and some with Euclidean distance 5 and graph distance 7. The function is monotonically decreasing in any case, hence exactly the same constraint can not be given.

Triangular:

1	1
$2, \sqrt{3}$	2
$3, \sqrt{7}$	3
$4, \sqrt{13}, 2\sqrt{3}$	4
$\sqrt{19}, \sqrt{21}$	5
$2\sqrt{7}$	6

$$\ell(dist) = \begin{cases} 6 & 0 < dist \leq 1 \\ 5 & 1 < dist \leq 2 \\ 4 & 2 < dist \leq 3 \\ 3 \quad \text{if} & 3 < dist \leq 4 \\ 2 & 4 < dist \leq \sqrt{21} \\ 1 & \sqrt{21} < dist \leq 2\sqrt{7} \\ 0 & 2\sqrt{7} < dist \end{cases}$$

Square:

1	1
$2, \sqrt{2}$	2
$3, \sqrt{5}$	3
$4, \sqrt{10}, 2\sqrt{2}$	4
$\sqrt{13}, \sqrt{17}$	5
$2\sqrt{5}, 3\sqrt{2}$	6
5	7
$4\sqrt{2}$	8

$$\ell(dist) = \begin{cases} 8 & 0 < dist \leq 1 \\ 7 & 1 < dist \leq 2 \\ 6 & 2 < dist \leq 3 \\ 5 & 3 < dist \leq 4 \\ 4 \quad \text{if} & 4 < dist \leq \sqrt{17} \\ 3 & \sqrt{17} < dist \leq 2\sqrt{5} \\ 2 & 2\sqrt{5} < dist \leq 5 \\ 1 & 5 < dist \leq 4\sqrt{2} \\ 0 & 4\sqrt{2} < dist \end{cases}$$

Exceptions: There are some vertex pairs with Euclidean distance $2\sqrt{2}$ and graph distance 4, and some with Euclidean distance 3 and graph distance 3. And there also are some vertex pairs with Euclidean distance $\sqrt{13}$ and graph distance 5, and some with Euclidean distance 4 and graph distance 4. The same is true as above.

Of course, in our study only the graph distances 1,2 and 3 were needed, the other distances would play a role if longer constraints were took.

Lattice	$s\ell_2(dist)$	$L(2,1)$	$s\ell_3(dist)$	$L(3,2,1)$
Hexagonal	5 (0.61 s)	5 (0.08 s)	9 (1.59 s)	9 (0.29 s)
Triangular	8 (7.94 s)	8 (0.34 s)	18 (1170.4 s)	18 (167.29 s)
Square	6 (6.51 s)	6 (0.2 s)	11 (68.91 s)	11 (0.68 s)

Although the conditions are the same for each vertex pair, the calculations are still much slower. But it should be noted that these running times compared to the ones with the linear functions are not that long.

1.3.3 Improving the formulation

Experience has shown that the precision has a high price, namely the time required for the calculations is significantly longer. We have made a few attempts to make the calculations faster. These are detailed below. In the followings the linear functions will be used for the comparisons.

Reducing the value of M A good value for M is clearly not smaller than the value that CPLEX would find for $\lambda + j_1$. But with this property it should be as small as possible. Since the bigger the value of M is, the larger the size of the problem. The fact is, that we have more inequalities, in this point the problem should be more complex. But in the other side, these inequalities can help in the process of Branch-and-Bound, that can make the calculations faster. So, it should be tried to see, whether the modification is effective in some cases. For this purpose, we must be able to estimate the results. A correct estimation of the value multiplied by a constant is already useful (if it is smaller than the original M-value). Since the exact values from the classical model are available, it is easy to get good values for the Euclidean model. In the case $\ell_2(dist) = 3 - dist$ we took the value $M_2^* = 2 \cdot \lambda_{L(2,1)}$, while $M_3^* = 3 \cdot \lambda_{L(3,2,1)}$ in the case $\ell_3(dist) = 4 - dist$. We got the following results:

Lattice	$\ell_2 = 3 - dist$, M_2^*	$\ell_2 = 3 - dist$, M
Hexagonal	6.41699 (1.82 s)	6.41699 (1.69 s)
Triangular	9.78029 (8.88 s)	9.78029 (10.96 s)
Square	8.63494 (7.25 s)	8.63494 (4.92 s)

Lattice	$\ell_3 = 4 - dist$, M_3^*	$\ell_3 = 4 - dist$, M
Hexagonal	16.6243 (6651.87 s)	16.6243 (1838.73 s)
Triangular	21.9068 (59064.01 s)	21.9068 (103883.6 s)
Square	19.9067 (21222.8 s)	19.9067 (27793.8 s)

One can see, that the modification works fairly well at the triangular lattice, but rather poorly at the hexagonals. We have acceleration, respectively slowdown, in the cases $\ell_2(dist)$ and $\ell_3(dist)$, as well. While the situations at the square lattice are not the same.

Limitation of λ Another approach is to include some good upper and lower bounds for λ in the formulation. In our case, of course, the proper upper bound is in question. We used the algorithm of binary search for determining it. The idea is to contract the lower and the upper bound as long as the gap between them (upper bound minus lower bound) is small enough. And the optimization may start only after this process with the received limits. We took the value received from CPLEX in the classical model for the initial lower bound, while we multiplicated it by 2 in the case of $\ell_2(dist)$, and by 3 in the case of $\ell_3(dist)$, for the initial upper bound. In each iteration either the lower bound is attempted to move upward or the upper bound down. There is an admissibility problem solved each time. We set the required gap to 0.5. Our results are shown in the following table (only the running times are indicated, since the outcomes are the same in both cases and as above):

Lattice	$\ell_2(dist) = 3 - dist$, new (s)	$\ell_2(dist) = 3 - dist$, old (s)
Hexagonal	5.18+21.59=26.77	1.69
Triangular	105.8+253.17=358.97	10.96
Square	33.99+55.26=89.25	4.92

Lattice	$\ell_3(dist) = 4 - dist$, new (s)	$\ell_3(dist) = 4 - dist$, old (s)
Hexagonal	118061+139576=257637	1838.73
Triangular	—	103883
Square	—	27793.8

The first addend in the sum is the running time, needed for the calculation of the bounds, while the second one is the running time, needed for the optimization, itself. In contrary to the expectations, the method did not improve the program, and even slowed it. This is probably due to the NP-hardness of the testing of admissibility of the Channel Assignment Problem. And in addition, after the reduction of the feasible set, it is difficult to cut off some branches in the search tree (Bounding). Therefore the optimum search takes longer time than previously.

Changing the order of the vertices In the optimum search algorithm of the CPLEX also the order of the vertices playes a role. Therefore one may want to try whether a new order would result any improvement. We considered reasonable to set up an order in which the successive vertices are not close to each other in the lattices. For this we changed the numbering according to the table.

$$\text{Square: } \begin{array}{ccc} 1 & \rightarrow & 1 \\ 2 & \rightarrow & 23 \\ 3 & \rightarrow & 20 \\ 4 & \rightarrow & 12 \\ 5 & \rightarrow & 9 \\ 6 & \rightarrow & 6 \\ 7 & \rightarrow & 3 \\ 8 & \rightarrow & 25 \\ 9 & \rightarrow & 17 \\ 10 & \rightarrow & 14 \\ 11 & \rightarrow & 11 \\ 12 & \rightarrow & 8 \\ 13 & \rightarrow & 5 \\ 14 & \rightarrow & 22 \\ 15 & \rightarrow & 19 \\ 16 & \rightarrow & 16 \\ 17 & \rightarrow & 13 \\ 18 & \rightarrow & 10 \\ 19 & \rightarrow & 2 \\ 20 & \rightarrow & 24 \\ 21 & \rightarrow & 21 \\ 22 & \rightarrow & 18 \\ 23 & \rightarrow & 15 \\ 24 & \rightarrow & 7 \\ 25 & \rightarrow & 4 \end{array} \text{, Triangular: } \begin{array}{ccc} 1 & \rightarrow & 1 \\ 2 & \rightarrow & 6 \\ 3 & \rightarrow & 11 \\ 4 & \rightarrow & 16 \\ 5 & \rightarrow & 21 \\ 6 & \rightarrow & 13 \\ 7 & \rightarrow & 18 \\ 8 & \rightarrow & 3 \\ 9 & \rightarrow & 8 \\ 10 & \rightarrow & 5 \\ 11 & \rightarrow & 10 \\ 12 & \rightarrow & 15 \\ 13 & \rightarrow & 20 \\ 14 & \rightarrow & 23 \\ 15 & \rightarrow & 17 \\ 16 & \rightarrow & 2 \\ 17 & \rightarrow & 7 \\ 18 & \rightarrow & 12 \\ 19 & \rightarrow & 9 \\ 20 & \rightarrow & 14 \\ 21 & \rightarrow & 19 \\ 22 & \rightarrow & 22 \\ 23 & \rightarrow & 4 \end{array} \text{, Hexagonal: } \begin{array}{ccc} 1 & \rightarrow & 1 \\ 2 & \rightarrow & 4 \\ 3 & \rightarrow & 16 \\ 4 & \rightarrow & 20 \\ 5 & \rightarrow & 24 \\ 6 & \rightarrow & 22 \\ 7 & \rightarrow & 14 \\ 8 & \rightarrow & 18 \\ 9 & \rightarrow & 9 \\ 10 & \rightarrow & 11 \\ 11 & \rightarrow & 8 \\ 12 & \rightarrow & 12 \\ 13 & \rightarrow & 3 \\ 14 & \rightarrow & 5 \\ 15 & \rightarrow & 2 \\ 16 & \rightarrow & 6 \\ 17 & \rightarrow & 19 \\ 18 & \rightarrow & 23 \\ 19 & \rightarrow & 15 \\ 20 & \rightarrow & 13 \\ 21 & \rightarrow & 17 \\ 22 & \rightarrow & 21 \\ 23 & \rightarrow & 7 \\ 24 & \rightarrow & 10 \end{array}$$

In practice, we only had to shuffle the rows and columns of the distance matrix properly. Unfortunately, in the end no improvement was observed.

Adding new inequalities to the formulation

Valid inequalities: The structure of valid inequalities is the same as that of the other inequalities in the formulation. So, in a linear programming formulation the left side is a linear combination of the variables, while the right side is an integer/real number. In opposite to the defining inequalities, the formulation is right also without them. But every solution of the input problem satisfies them. The point why valid inequalities are useful, is that they can make the calculations faster, since they express the additional dependencies between the variables. We used inequalities based on the sums of the labels in some subgraphs. Thereby the problem of minimizing the sum of the labels should be solved separately for the subgraphs. Let λ' be the result of such a problem for some subgraph G'. The additional inequality corresponding to G' is then $\sum_{v\in V(G')} c(v) \geq \lambda'$. Such inequalities are added to the formulation for each subgraph isomorphic to G'. Our calculations are presented below.

Stars as subgraphs Our goal is to determine an inequality, in which on the left side is the sum of the labels of the vertices of a star. For this one should give a lower bound for the right side. So the task is to count the minimum sum of labels in a star. A star contains a central vertex with deg number of neighbors, and these neighbors are pairwise non-adjacent. If the number of constraints is at least 2, the labels in the star are pairwise different. The labals are from a label spectrum which includes the natural numbers from 0 to λ. In this spectrum $deg + 1$ numbers should be chosen so that the gap between the label of the central vertex and the label of any other vertex is at least j_1, and the gap between the labels of any two non-central vertices is at least j_2.

No gap is larger than necessary, because in this case the sum could be done smaller by taking the gap smaller, while the constraint would continue to complete. So, with the necessary gaps there are 3 essentially different possibilities: The central vertex has the smallest label or the largest label or an intermediate one. In these three cases one gets the sum of the values by the followings:

1. The central label is the smallest one, the labels are 0, j_1, $j_1 + j_2$, $j_1 + 2j_2$, $j_1 + 3j_2$,..., $j_1 + (deg - 1)j_2$. The sum is $deg \cdot j_1 + \sum_{i=1}^{deg-1} i \cdot j_2$.

2. The central label is the greatest one, the labels are 0, j_2, $2j_2$,..., $(deg-1)j_2$, $(deg - 1)j_2 + j_1$. The sum is $(\sum_{i=1}^{deg-1} i \cdot j_2) + (deg - 1)j_2 + j_1$.

3. The central label is an intermediate one, the labels are: 0, j_2, $2j_2$,..., $(k - 1)j_2$, $(k - 1)j_2 + j_1$, $(k - 1)j_2 + 2j_1$, $k \cdot j_2 + 2j_1$,..., $(deg - 2)j_2 + 2j_1$. (Here, the central vertex is in the $(k + 1)^{th}$ position.) The sum is $(\sum_{i=1}^{deg-2} i \cdot j_2) + 2(k - 1)j_2 + (2(deg - k) + 1) \cdot j_1$.

Since $j_1 \geq j_2$, the first sum is bigger than the second one. The comparison of the second and third sums: $(2deg - 2k + 1)j_1 \geq j_1 + (2deg - 2k)j_2 \Longrightarrow 2(k - 1)j_2 + (2deg - 2k + 1)j_1 \geq j_1 + (2deg - 2)j_2 = 2(deg - 1)j_2 + j_1$. So, the minimal sum is $\sum_{i=1}^{deg-1} i \cdot j_2 + (deg-1)j_2 + j_1 = \frac{deg\cdot(deg-1)}{2} \cdot j_2 + (deg-1)j_2 + j_1 = \frac{(deg+2)(deg-1)}{2} \cdot j_2 + j_1$.

A graph contains n number of stars, namely each vertex as central one with its neighbors form a star. Using the above formula the following inequality can be written for any vertex k:

$c(k) + \sum_{i=\text{each neighbor of the vertex } k} c(i) \geq \frac{(deg(k)+2)(deg(k)-1)}{2} \cdot j_2 + j_1$.

In the following these inequalities are called the star-inequalities.

Sublattices As the star-inequalities have been introduced into the formulation for general graphs, some further ones can be introduced into the formulation for each graph classes, depending on what kind of subgraphs they contain. Since the currently investigated lattice graphs are fairly uniform, one can easily define some subgraphs that occur often. Practical these are small lattices. Also in this case the inequalities relate to the minimum sum of the vertex labels. We considered the triangels with side length 1,2 and 3, hexagons with side length 1 and 2, trapeze as the half part of a hexagon and squares with side length 1,2 and

3, and small square lattices with 4,9 nodes, as well. The followig table containes the optimum values of the sums $\sum_{v\in V(G')} c(v)$ for each studied sublattice.

Sublattice	$\ell_2(dist) = 3 - dist$	$\ell_3(dist) = 4 - dist$
Hexagons with side length 1	16.6077	31.6077
Hexagons with side length 2	3	10.8231
Trapeze with side length 1	7.0718	13.0718
Triangle with side length 1	6	9
Triangle with side length 2	3	6
Triangle with side length 3	0	3
Square 1×1 (4 nodes)	10.3431	16.3431
Square 2×2 (4 nodes)	2.68629	8.68629
Square 3×3 (4 nodes)	0	2
Squarelattice 2×2 (9 nodes)	29.1922	59.4033
Squarelattice 1×2 (6 nodes)	16.2273	30.283

A summary of the best results is presented in the following tables:

Hexagonal lattice	$\ell_2(dist) = 3 - dist$	$\ell_3(dist) = 4 - dist$
Without any subgraph-inequality	1.69 s	1838.73 s
With the subgraph-inequalities		
Hexagons with side length 1	1.37 s	1588.7 s
Hexagons with side length 1 and 2	1.75 s	1641.71 s
Trapeze	1.77 s	1529.24 s

The approach provides no improvement in running time.

Triangular lattice	$\ell_2(dist) = 3 - dist$	$\ell_3(dist) = 4 - dist$
Without any subgraph-inequality	10.96 s	103883 s
With the subgraph-inequalities		
Triangels with side length 1	12.59 s	46933 s
Triangels with side length 1,2 and 3	10.02 s	92228.1 s
Hexagons with side length 1	11.9 s	42412 s

An improvement in the running times is observed, but the calculations still need a lot of time, and with the introduction of the additional constraints even more storage space.

Square lattice	$\ell_2(dist) = 3 - dist$	$\ell_3(dist) = 4 - dist$
Without any subgraph-inequality	4.92 s	27793.8 s
With the subgraph-inequalities		
Squares with side length 1	12.76 s	37945.53 s
Squares with side length 1,2 and 3 *	7.11 s	19619.38 s
Tetragon with side length 1×2	6.22 s	28612.78 s

* inequalities added only for non-overlapping squares (substantially less than existing)

The approach brings in turn no significant improvement in the running times.