1 HPA*

Botea et al. report on HPA*, a hierarchical technique for fast path planning [**?**] for a limited class of terrains which we refer to as *grid-maps* (we will return to the issue of limitation after we present the method). These terrains have unit edge weights (representing distance between adjacent nodes) and have obstacles represented by nodes that cannot be visited at any cost (effectively, the nodes representing obstacles are removed from the graph).

The technique builds a hierarchy of graphs representing the connectivity of the original; this is done in a pre-processing step. The bottom level (level 0) of the hierarchy is the original grid-map. Nodes in the first level are way-points chosen from the original, and on every level higher than the first level, a subset of the way-points represent the graph immediately below it in the hierachy.

Search in the hierarchy has three steps. First, the end-points of the desired path are inserted into the hierarchy. Second, a path is found through the top level of the hierarchy. Third, the path is refined by working from the top level down to the original grid-map. The costs of pre-processing can be amortized over many search queries in the hierarchy, but each search query invokes the over-head of inserting the start and end-points into the hierarchy.

1.1 Constructing the hierarchy

To construct the first level of the hierarchy, the original grid-map is partitioned into into square clusters of $b \times b$ nodes, where b is a user-defined parameter. We will refer to this choice as the *initial cluster size*. Way-points are chosen from the nodes on the boundaries of the cluster, representing *entrances* into (end exits from) the cluster. An entrance is defined as a maximal sequence of nodes containing no obstacles on the border of a cluster, which has an

The entrance concept is much simpler than the definition may suggest: any node node on the border of a cluster that is not an obstacle and is not adjacent to an obstacle in the adjacent cluster is a way into the cluster, and any sequence of such nodes represents a single opportunity for entering the cluster. Depending on the arrangement of obstacles and choice of initial cluster size, there may be several entrances to a cluster along a given boundary.

If the entrance is longer than a given threshhold (the authors use 6 nodes), two nodes on either end of the entrance are chosen as way-points; if the entrance is smaller, a node in the middle of the sequence is chosen. Each node chosen to be a way-point for a cluster has a symmetric way-point on the adjacent cluster.

Once the clusters and way-points have been established in the original grid-map, the levels of the hierarchical structure is constructed as follows, and is true for all levels. The way-points chosen at level i are the nodes in the graph at level $i + 1$. Two kinds of edges are created at levels 1 and higher: inter-edges and intra-edges. An inter-edge is placed between symmetric way-points at the cluster boundaries of the level below; the weight of this edge is the weight of the edge it represents in the original grid. Intra-edges connect pairs of way-points within a given cluster: an intra-edge is created if there is a path between them using only nodes in the cluster, and it is weighted by the cost of the best of all such paths.

The user also defines m, the number of levels to be constructed in the hierarchy. The first level is constructed by partitioning the original grid-map into $b \times b$ clusters, and identifying entrances, etc. Clusters at level $i > 1$ are composed by taking the union of a number of clusters at level $i - 1$. The authors use 4 clusters at level $i - 1$, arranged in a 2×2 pattern, as a cluster at level i. For example, if the user defined the cluster size $b = 10$, then a first level cluster represents a 10×10 area in the original grid; a 2nd level cluster represents a 20×20 area. The number of clusters at level $i + 1$ is one quarter the number at level i (ignoring the slight complication that n may not be an even multiple of b nodes).

Because some of the boundaries between level 1 clusters persist at level 2, the level 2 way-points consist of the set of level 1 way-points that are on level 2 boundaries. Likewise, level 2 inter-edges are those level 1 inter-edges that cross level 2 boundaries. New intra-edges are placed between pairs of way-points at level 2. As before, an intra-edge is placed if there's a path between them within the bigger cluster, and it is weighted by the cost of the best such path. Further levels of the hierarchy are constructed in the same way, by combining clusters at the immediately lower level, identifying the way-points that are on the boundaries between the larger clusters, and constructing new intra-edges.

Pre-processing costs The intention of the technique is to produce a hierarchy of smaller graphs, to make search much more efficient. However, the technique does not guarantee that the graphs get smaller. During the pre-processing, as clusters get larger, it is possible for the number of waypoints in a cluster to increase, and the number of intra-edges as well. Because each pair of entrances to a cluster is joined by an edge if there is a path between them contained entirely in the cluster, the number of intra-edges in the cluster is quadratic (worst case) in the number of way-points contained in the cluster.

The actual number of intra-edges depends on the structure of the original grid-map. If the grid-map represents a floor-plan with limited connectivity between rooms, and few open regions, then the number of way-points may be very small (maybe only two way-points in the entire cluster, if the cluster contains only a long hallway, or a room with 2 doors). Thus the number of intra-edges will be small as well. If the cluster contains several hallways or tunnels, but they are not connected within the cluster, then the number of intra-edges can also be much smaller than a complete graph. This is the kind of graph that the authors chose to use for their experiments.

However, if the grid-map represents open areas, with only a few obstacles (rocks, trees, etc), then almost every way-point has a path to every other way-point in the same cluster, and the number of intra-edges can be very high in high levels of the hierarchy.

Consider, for example, a grid-map with no obstacles at all. The degree of each node is a small constant (maximum of 4 or 8, depending on the connectivity). The higher levels of the HPA* hierarchy for an empty gridmap are different. We will show that each level in the hierarchy has $O(n^2/b^2)$ intra-edges, i.e., linear in the number of nodes in the original graph. This is due to the following fact: while the total number of nodes roughly halves at each level up the hierarchy in a grid-map with few obstacles, the number of way-points into each cluster roughly doubles. These ideas are presented slightly more formally here.

Result 1.1 *In a grid-map without obstacles, the number of nodes in each cluster at level* $i + 1$ *is twice the number of nodes in a cluster at level* i*, for* $i > 1$*.*

Proof: Without obstacles, the number of nodes depends on the size of the initial cluster, since there are no obstacles to work around. If $b \leq 6$ then $k = 1$ way-points per side are used. If $b > 6$ then the number of way-points depends on how they are placed. There are no more than $k = 2$ waypoints per side. In any case, k is a small constant.

For each level $i > 1$, the new clusters are composed by taking the union of 2×2 clusters at level i. If there are k_i nodes per side at level i, for a total of $4k_i$ nodes per cluster at level i, then there are $2k_i$ nodes per side at level $i + 1$, for a total of $8k_i$ nodes per cluster at level $i + 1$. QED

The claim and the proof assume that the placement of way-points is uniform and regular, and in practice, this is true. The claim ignores boundary effects that arise when clusters are composed using unequal sized clusters, which arise in practical situations when the dimensions of the opriginal grid-map are not an even multiple of b.

Result 1.2 *In an grid-map without obstacles, the number of nodes at level* $i + 1$ *is half the number of nodes at level* i*, for* $i > 1$*.*

Proof: Since the total number of nodes per cluster increases by a factor of 2, and the number of cluster is reduced by a factor of 4, the total number of nodes is reduced by a factor of 2. QED

Result 1.3 *In a grid-map without obstacles, the number of intra-edges at each level is* $O(n^2/b^2)$.

Proof: There are $k2^{i+1}$ nodes per cluster at level $i > 0$. There are $n^2b^{-2}2^{-2i+2}$ clusters at level i. There are $k^2 2^{2i+1} - k2^i$ intra-edges per cluster, assuming complete connectivity within the cluster. There are $n^2b^{-2}2^{-2i+2}(k^2 2^{2i+1} - k2^i) = O(n^2/b^2)$ intra-edges at level i. QED

Note that this result also makes the simplifying assumption that n is big enough to allow any given number of levels in a hierarchy. This of course is not true, but it is true for all levels in which most clusters at level $i + 1$ are made up of 4 roughly equal sized clusters at level i.

Thus, on grid-maps with limited connectivity, the number of nodes and edges decrease with each level in the hierarchy. However, on more general terrains, the number of nodes decreases, but the number of edges remains roughly constant.

The user must provide two parameters, b, the cluster size, and m the height of the hierarchy. These also affect the pre-processing costs,as well as the on-line search costs. The choice of b and m can be tuned to the features of the graph; for example, b could be related to the typical size of a room, and m would be just large enough to make the top level search negligible in cost. In any case, large b implies a large reduction in the number of nodes at level 1 of the hierarchy; small m ensures that the number of edges is small in the worst case. However, b should not be too large, because the clusters must be searched repeatedly during pre-processing, and during on-line searches.

1.2 On-line search

Search in the hierarchy consists of finding a best path at the highest level of the hierarchy, and then refining the path through lower levels of the hierarchy. Since a path-finding exercise may require a path between two nodes that are not identified as way-points, the technique inserts the start and goal node into the hierarchy as temporary way-points. Each cluster on every level representing an area containing the start node has a new node inserted, and new intra-edges are created between the start node and the way-points for the cluster; the same is done for the goal node. Note that this entails a number of searches at each level in the hierarchy. For low levels of the hierarchy, the costs are small, but the number of nodes per cluster doubles at every level up the hierarchy.

A path through the highest level is basically a sequence of waypoints at cluster entrances, tracing across intra-edges and inter-edges. The optimal path at this level represents a good quality path, because the cost of each intra-edge is exact; deviations from optimal quality are the result of forcing paths through waypoints (and this can be alleviated by post-processing of the path; see below).

A high-level path can be refined by replacing any intra-edge on the path with the (best) sequence of nodes in the lower level cluster that the intra-node represented. If space is available to store these paths, no search is required to do this; otherwise, a limited search is conducted to identify this path. Any search method can be used; however, because the cost of the optimal path is known, it may be worthwhile to use this information to prune the search space, especially if the cluster size is relatively large: discard any successors whose f-cost is higher than the cost of the intra-edge.

The path is fully refined when all intra-edges are replaced with edges of the original grid. The order in which this is done does not matter; an abstract path can be refined as the path-finding agent proceeds along the path, saving effort if the whole path is not needed (e.g., if the agent receives a new target before reaching the current one).

The path may not be optimal because it has been forced to pass through way-points chosen to represent entrances between regions in the original grid-map. This can result in paths that are forced to deviate from a more direct path just to pass through a way-point. The authors suggest using a path smoothing operation to help correct for this effect. The proposed smoothing procedure looks to insert straight-line paths between nodes on the refined path to replace such deviations.

Because the way-points at level $i + 1$ are way-points in the level below, and because the intra-edges represent the cost of paths between the way-points, the path cost at every level is the same; a refined path between merely replaces an edge between way-points, but is guaranteed to have the same cost.

1.3 Grid-maps v. terrains

It is fair to say that HPA* applies to the class of grid-maps because, as we have described, the choice of way-points does not consider the cost of travelling through the way-points. In a grid-map, every non-obstacle is potentially equal with respect to costs. Thus the paths may be forced through a way-point, but in a grid-map, the edge costs are never high enough to make this a very bad decision. We note that in a terrain, where edges may vary in cost a great deal, it could be very expensive to force a path through a way-point chosen without regard to the edge costs into it.

We also note that the path smoothing technique proposed for HPA* does not consider edge costs either. While a more direct route is always preferred in a grid-map, the result of the path smoothing operation proposed by the authors may may force the path through a region of high cost edges in a terrain, because it is more direct.

However, these points do not prevent HPA* from being applied to general terrains. The choice of way-points would be the same choice as in a simple grid-map with no obstacles, but the intra-edges would be computed using the edge weights of the terrain. Thus the hierarchy will accurately represent the cost of paths between aribtrarily chosen way-points.

2 Experiments

We applied HPA* [**?**] to the four mazes, the four Freeciv terrains, and the six representative Brodatz terrains. The purpose of our investigation was to evaluate HPA* in terms of pre-processing costs, path quality, and on-line search costs.

In order to keep runtimes reasonable, we performed preliminary experiments on a variety of terrain sizes, and we present detailed results for the largest size that each aspect of our investigation allowed. Reasonable runtimes were a concern because of the implementation of HPA* provided by the authors of the technique: substantial overhead was needed to preprocess a grid-map, above the costs due to the basic construction of the hierarchy. Search costs (as measured by cpu time) during HPA* preprocessing accounted for only about two thirds of the total time needed to complete the preprocessing; much of this hidden cost is due to inefficiencies that were acceptable for the small grid-maps used in [**?**], and considering the code was not intended for production use. The implementation also suffered from an inefficient use of space, which incurred some limitation to the size of terrain, depending on the nature of the investigation. We emphasize that these were practical limitations on our experimentation, but not limitations of the method in principle.

2.1 Experiments on grid-maps

The HPA* technique implicitly assumes a grid-map as input, i.e., a terrain with obstacles, but edge weights representing unit distances between vertices in the terrain. We used 4 mazes in our experiments, which are examples sof this class of terrain.

The Freeciv terrains and the representative Brodatz terrains are not grid-maps, but general terrains. However, it is possible to derive a grid-map from a general terrain by threshholding the edge weights: an obstacle is placed whenever all the edge-weights into a node in the terrain exceeds a given threshhold value. In all the terrains we derived from images, this can be done simply by threshholding the pixel values. The Freeciv terrains can also be handled this way.

For each of the Brodatz terrains, a threshold value was chosen so that the obstacle density (sometimes referred to as ϕ or just `phi` in text) was as close to 10%, 20% and 30% as possible. The nodes in each terrain were then converted to a node or an obstacle in the grid-map, depending on whether the pixel was lower or higher than the threshhold. The Freeciv terrains did not have much variance in the edge weights, so only the 10% obstacle density was used.

The Brodatz terrains are 729×729 in dimension, and the implementation of HPA* required between 25 and 40 minutes to process them. For this reason, we scaled these terrains down, ranging from 243×243 up to 729×729 by increments of 81 nodes per side. Thus, there were 18 Brodatz grid-maps for each graph dimension. The Freeciv terrains and the 4 mazes had dimension 243×243.

Our experimental method consisted of constructing a hierarchy for each grid-map, running 1000 random path queries in the hierarchy, and collecting data from these trials.

2.1.1 Pre-preprocessing costs

We measure the size of the hierarchy constructed by HPA* in terms of the number of nodes and edges at each level. We used an initial cluster size of $b = 10$ and built $m = 2$ levels in the hierarchy. We used these values because they were the values used in the initial report [**?**].

Table 1 shows the storage costs of HPA* for the mazes and the Freeciv grid-maps. We observe that the number of nodes in the first level is a small fraction of the number of nodes in the original grid-map, and that the number of nodes decreases at level 2. A decrease in the number of inter-edges as is also evident (which is implied by the decrease in the number of nodes).

However, the number of intra-edges increases in 5 of the 7 examples. In the small maze, for example, the number of intra-edges increases from 9977 at level 1 to 11255 at level 2. The number of clusters drops from 25^2 in level 1 to 13^2 in level 2, giving an increase in intra-edges from about 16 per cluster to about 66 per cluster. In the Freeciv terrains, the number of nodes is in the first level of the hierarchy is somewhat larger than for the mazes, but the number of intra-edges is substantially larger, and increases at level 2. For the Freeciv 1 grid-map, for example, there are about 38 intra-edges per cluster at level 1, and 155 at level 2.

Note also the size of the graphs in the HPA* hierarchy for the large maze. This maze is somewhat larger than the other mazes, in terms of dimension, but the number of nodes and intra-edges is substantially smaller at all levels than any of the other mazes. This is because the large maze has a structure that limits the connectivity of way-points for each cluster: there are about 2 intra-edges per cluster at level 1, and about 4 intra-edges per cluster at level 2.

In all of these example grid-maps, the number of nodes and inter-edges decreases by half from level 1 to level 2, as expected, and the number of intra-edges increases slightly in 5 of them. For the Freeciv terrains, the number of intra-edges is larger than in the mazes, and is a larger fraction of the number of edges in the original graph.

We applied the HPA* technique to the 729×729 Brodatz grid-maps as described above, again with $b = 10$ and $m = 2$. The results are summarized in Tables 2.

These grid-maps are much larger than the examples from Table 1, but the same patterns are evident. On average, the number of nodes at level 1 is 7.5% of the nodes in the original graph, and the number of nodes

	Original		level 1			level 2		
Map	nodes	edges	nodes	inter	intra	nodes	inter	intra
small maze	55437	217585	3775	2137	9977	1863	1062	11255
large maze	65346	197104	4356	2178	2346	2230	1115	1266
russian dolls	53545	204468	4090	2277	9006	2104	1187	8598
russian quad	51753	194612	3940	2134	7955	2172	1188	8189
Freeciv 1	50251	173473	5580	2946	23561	2805	1483	26183
Freeciv 2	50274	173598	5555	2926	23291	2765	1454	25024
Freeciv 3	49516	169111	5515	2906	22971	2751	1454	24750

Table 1: Space used by HPA* on the mazes and Freeciv terrains. The large maze is 340×340 in size, and all the other graphs are 243×243.

		Original		level 1			level 2		
Map	ϕ	nodes	edges	nodes	inter	intra	nodes	inter	intra
D24	0.1	481013	1864426	35811	19875	100673	17932	9996	124405
D3	0.1	475646	1844010	35133	19512	93777	17558	9779	109855
D40	0.1	459411	1775647	33948	18918	92961	17004	9487	113368
D44	0.1	481634	1915789	33631	19127	98073	16745	9508	122662
D49	0.1	481066	1854953	35291	19495	85440	17469	9643	88226
D76	0.1	474050	1856200	34228	19184	95007	17084	9573	114332
D24	0.2	427432	1618818	33085	18113	82415	16608	9136	96221
D3	0.2	424133	1618021	32160	17713	77274	16083	8888	80388
D40	0.2	427572	1640006	32021	17785	84044	16005	8923	100600
D44	0.2	435771	1707252	31871	17841	94461	15837	8876	115981
D49	0.2	422332	1593174	32068	17408	64351	16099	8737	55488
D76	0.2	427184	1653089	31547	17506	82889	15781	8757	95959
D24	0.3	367780	1360293	29543	15987	64488	14878	8084	67466
D3	0.3	367130	1381124	28225	15452	62487	14169	7790	58863
D40	0.3	345306	1304637	26342	14536	63552	13223	7337	73277
D44	0.3	377849	1409469	30461	16517	84550	15131	8248	98475
D49	0.3	366783	1366288	28495	15350	51367	14317	7713	41564
D76	0.3	373747	1424636	28231	15539	69185	14109	7784	76178

Table 2: Space used by HPA* on the Brodatz grid-maps. The grid-maps are 729×729 in size, and ϕ is the obstacle density. We applied HPA* using a cluster size of 10, and a 2 level hierarchy.

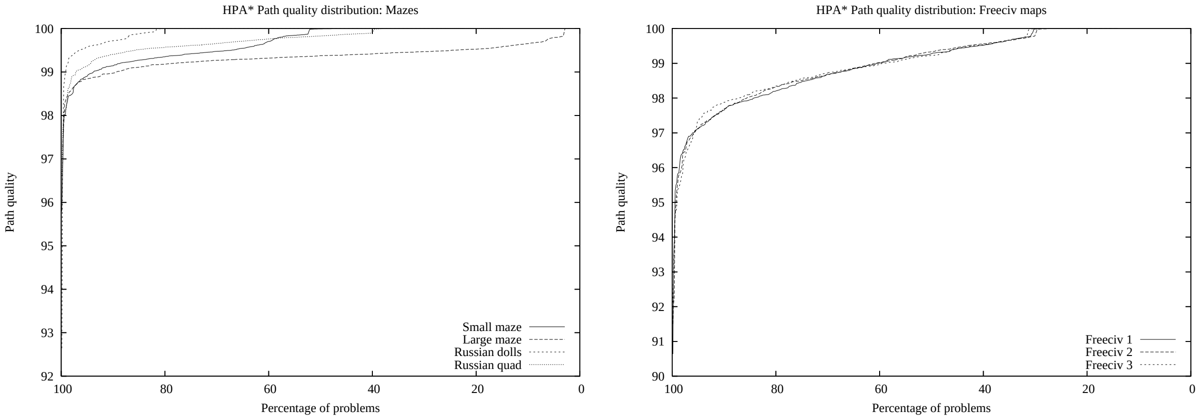

Figure 1: Cumulative distribution of path quality from applying HPA* to the maze terrains (left) and the Freeciv grid-maps (right).

at level 2 is half the number at level 1. For all but 3 examples, there are slightly more intra-edges at level 2 than at level 1. The average number of intra-edges at level 1 is 4.9% of the original, and at level 2, 5.5% of the original. The number of intra-edges increases from level 1 to level 2 in 15 of the 18 terrains, by 11% of the level 1 intra-edges, on average.

Clearly, HPA* cuts the size of the graph significantly at the first level, by partitioning into $b \times b$ clusters, but the number of intra-edges does not decrease with the height of the hierarchy. This is as predicted by our rough analysis. Because of this, short hierarchies may be preferred over hierarchies with many levels. In terms of storage, a large b is preferred, because it reduces the number of nodes at level 1 of the hierarchy.

2.1.2 On-line search performance

To assess on-line search performance, we used a set of 1000 pathing queries, generated by selecting end-points at random. To simplify the process, if either of the end-points chosen referred to the location of an obstacle, the trial was excluded from the sample, but no replacement was made.

To evaluate the results, we computed path quality for each path, i.e., the ratio of the optimal path to the path found by HPA*. This quantity is used because it is bounded between 0 and 1.

We applied HPA* directly to the mazes, and to the grid-maps derived from the 486×486 Brodatz grid-maps and the Freeciv grid-maps. We produced cumulative distributions for HPA* showing the frequency of path quality. Figure 1 show the distributions for the mazes and the Freeciv grid-maps. Table 3 gives quantitative results for the mazes, showing the path quality obtained by the best 75%, 90% 95% and 99% of the trial paths. The path quality is very good across these grid-maps, with at least 99.9% of the paths having a path quality of 95% or better.

The results for the Brodatz grid-maps are summarized in Tables 4 and 5. The results from the Freeciv grid-maps are also shown in Table 4, because they had roughly similar obstacle density. Figure 2 shows a typical path quality distribution for a single Brodatz terrain, for the three obstacle densities used, and the path quality distributions for all the Brodatz terrains for obstacle density $\phi = 0.2$.

The path quality in all these grid-maps is very good, achieving 95% with almost certainty. We observe that the Freeciv terrains had noticably lower frequency of achieving 99% path quality.

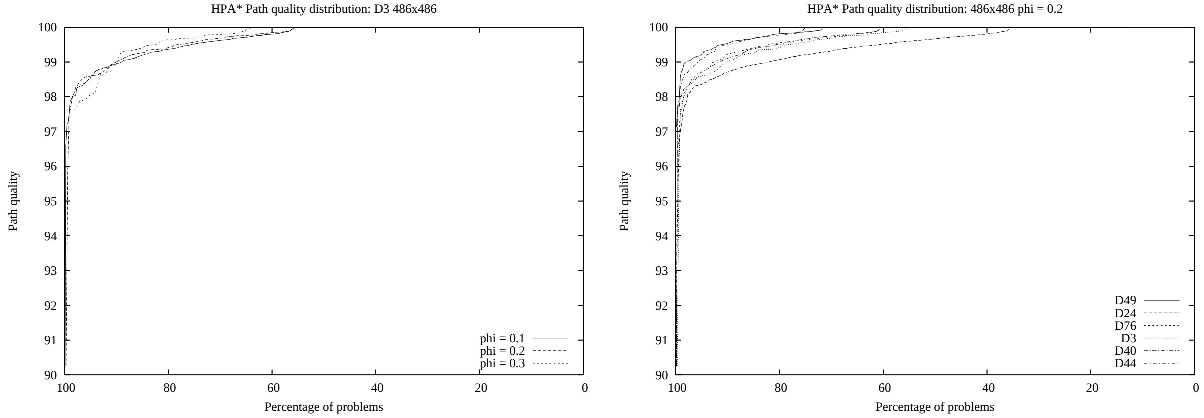

Figure 2: Cumulative distribution of path quality from applying HPA* to the one of the Brodatz grid-maps, D3, for three obstacle densities (left), and for all Brodatz terrains of obstacle density 0.2 (right).

Map	99%	95%	90%	75%
small maze	93.9	100	100	100
large maze	89.4	100	100	100
russian dolls	99.1	99.9	100	100
russian quad	97.3	100	100	100

Table 3: Cumulative distribution of path quality from applying HPA* to the maze terrains. These size of these terrains is 243×243.

Map	99%	95%	90%	75%
D49	97.9	99.9	100	100
D24	89.8	99.9	100	100
D76	97.2	99.7	100	100
D3	88.9	99.9	100	100
D40	95.3	99.9	100	100
D44	99.6	99.9	100	100
Freeciv 1	60.2	99.6	100	100
Freeciv 2	60.8	99.3	100	100
Freeciv 3	59.3	99.1	100	100
Freeciv 4	59.3	99.1	100	100

Table 4: Cumulative distribution of path quality from applying HPA* to the grid-maps derived from the representative Brodatz terrains, and the Freeciv grid-maps. The obstacle density of the grid-maps is 0.1.

	$\phi = 0.2$					$\phi = 0.3$			
Map	99%	95%	90%	75%		99%	95%	90%	75%
D49	97.8	100	100	100		98.6	100	100	100
D24	82.1	99.7	100	100		83.3	100	100	100
D76	92.6	99.8	100	100		92.9	100	100	100
D3	90.2	99.7	100	100		90.1	100	100	100
D40	91.6	99.8	100	100		88.9	100	100	100
D44	95.6	99.8	100	100		79.2	99.8	100	100

Table 5: Cumulative distribution of path quality from applying HPA* to the Brodatz grid-maps . The obstacle density of the grid-maps is 0.2 (left) and 0.3 (right).

2.2 On-line search costs

To measure the search costs for pathfinding with HPA* applied to grid-maps, we recorded the number of node expansions for each path. The number of node expansions is drastically less than needed by A*; Figure 3 shows a typical side-by-side comparison of the cost of searching through one of the Brodatz-derived grid-maps.

The A* data shows the quadratic behaviour expected from explanding an area around a path, and the number of expansions required by HPA* is well below A*. Figure 4 shows the HPA* results seaparately; note that the number of expansions increases faster than linearly, though the spread is relatively high. In Figure 5, we plot, for each path, the number of nodes expaded by HPA* against the number of expansions needed by A*. The data show a strong correlation, which, informally, seems just a little shy of linear. The data is nearly a linear relationship, and the slope of a line of best fit would indicate the approximate ratio of HPA* expansions to A* expansions. In the case of the graph shown, the "slope" is about $1/20$. The pronounced y-intercept is due to the overhead involved in inserting the start and end points for each path into the appropriate clusters in each level of the hierarchy.

Thus we can see that HPA* uses far fewer node expansions than A* for on-line search, and results in paths whose quality is very close to optimal almost all the time. Variations in topology do not seem to affect the path quality very much, but it does have a modest impact on search costs, which can vary anywhere from 10 to 20 times fewer than the search costs of A*. The search costs also seem to increase faster than linear when viewed as a function of path length. However, this is almost certainly due to the fact that the hierarchy was only 2 levels above the original gridmap, and a significant portion of the top level graph needed to be searched for longer paths.

2.3 Grid-maps v. terrains

We also investigated the use of grid-maps to approximate terrains. We recorded the paths obtained from the grid-maps, and determined the cost of the path through the original terrain.

For this part of our investigation, we used grid-maps derived from the Brodatz terrains sampled to 486×486. We ran 1000 pathfinding queries, on the Brodatz grid-maps, using obstacle densities 0.1, 0.2 and 0.3.

The results are quite different from the results of HPA* on grid-maps, presented above; see Table 6. A path quality of 90% is achieved very rarely in 4 of the six grid-maps, and a path quality of 75% is achieved less than 50% of the time in 4 of the 6 grid-maps.

Essentially, the conclusion we can draw from these tables is that a grid-map is not a very useful approximation for a terrain, and so methods for dealing specifically with terrains must be used.

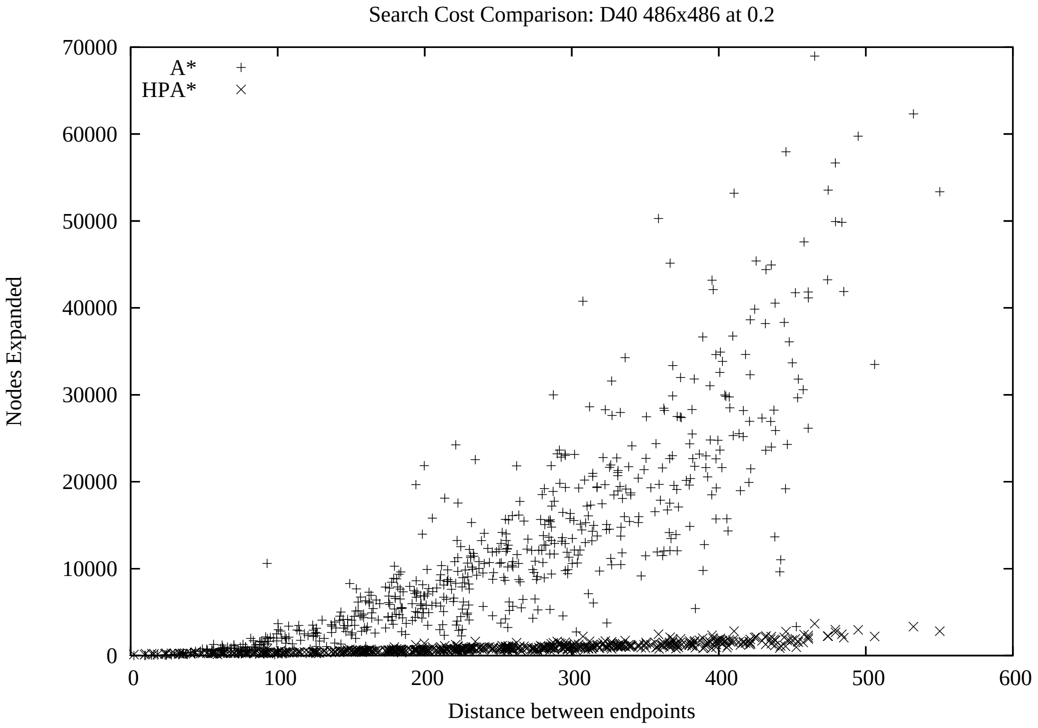

Figure 3: The number of expansions by A* and HPA*for Brodatz D3 grid-map, with obstacle density $\phi = 0.2$.

	$\phi = 0.1$					$\phi = 0.2$					$\phi = 0.3$			
Map	99%	95%	90%	75%		99%	95%	90%	75%		99%	95%	90%	75%
D49	0.3	0.7	1.1	3.1		0.5	0.9	1.9	7.2		0.6	1.0	2.4	20.2
D24	0.3	0.4	0.9	20.5		0.2	0.3	0.8	27.1		0.2	0.4	1.1	16.7
D76	0.6	2.0	9.7	56.5		0.3	1.1	5.8	46.5		0.4	0.6	2.0	22.3
D3	0.2	0.6	1.7	26.3		0.8	1.3	4.2	31.9		0.9	1.8	8.3	54.1
D40	2.0	7.7	23.9	86.6		1.7	6.0	23.0	85.2		1.2	4.6	16.7	75.6
D44	1.1	31.9	92.5	100		1.6	37.4	95.3	99.7		0.4	34.3	87.3	98.7

Table 6: Cumulative distribution of path quality from applying HPA* to the representative Brodatz terrains. The terrains were size 243×243 with three obstacle densities.

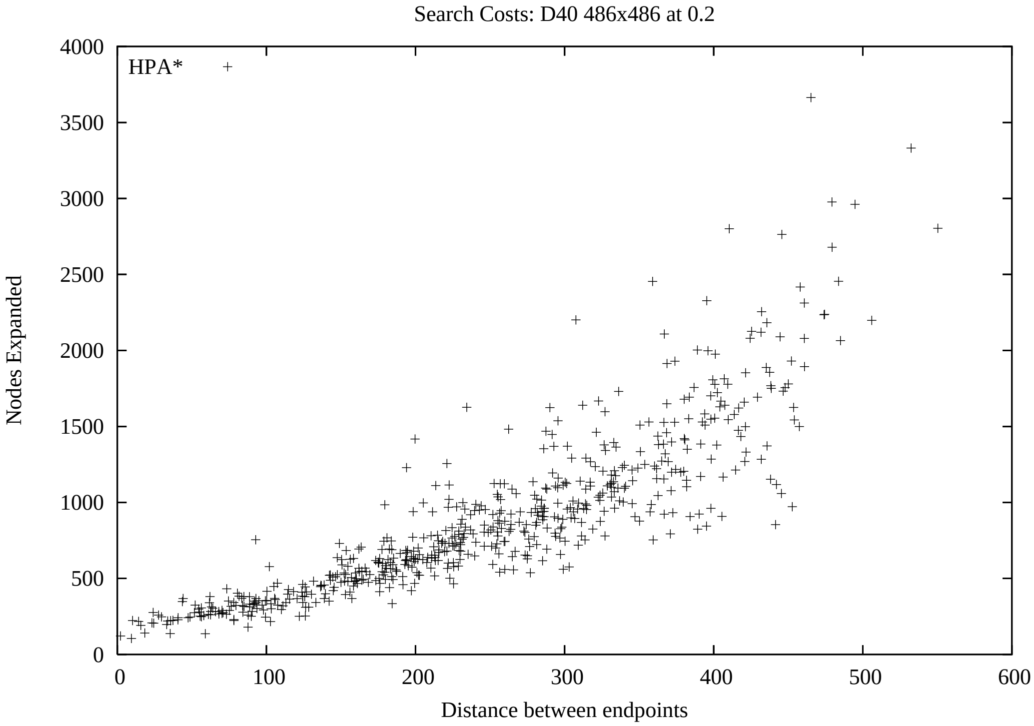

Figure 4: The number of expansions by HPA*on Brodatz D40 with $\phi = 0.2$.

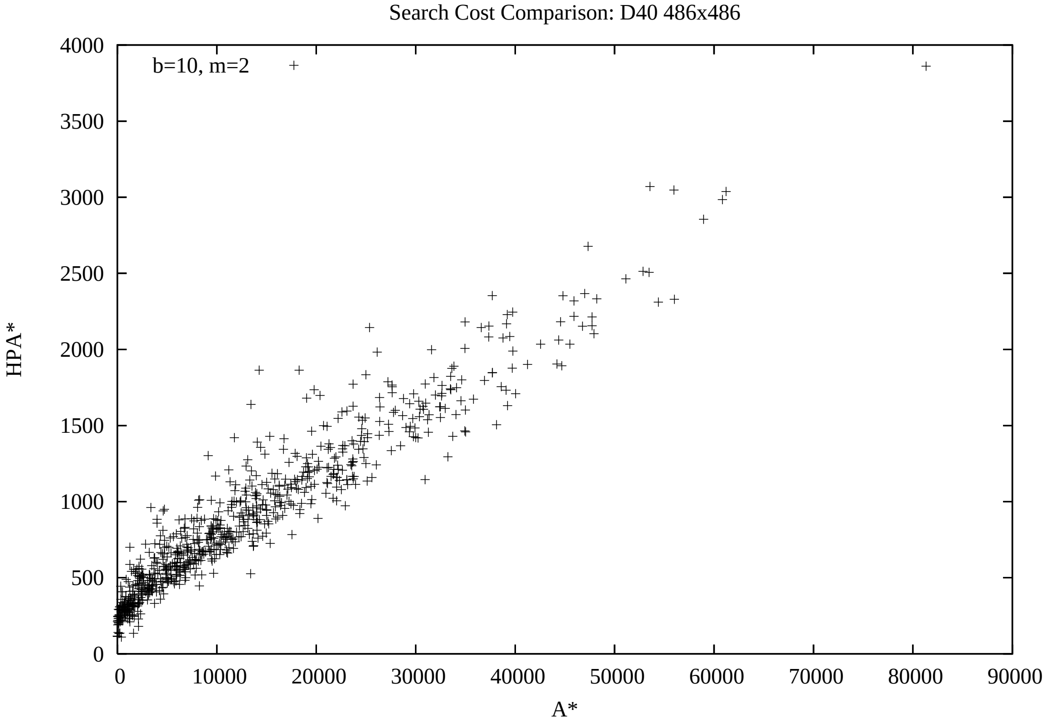

Figure 5: The number of expansions by HPA* plotted against the number of expansions needed by A*, for each path, for Brodatz D40 with $\phi = 0.2$.

	$b = 5$			$b = 10$		
Level	nodes	inter	intra	nodes	inter	intra
1	9355	4704	13820	4128	2352	11949
2	4653	2352	15836	2064	1176	14879
3	2351	1176	16372	1032	588	15055
4	1175	588	15812	516	294	13667
5	587	294	13960	—	—	—

Table 7: Space used by HPA* on terrains of size 243×243. Hierarchies were constucted up to level 4 for cluster size of $b = 10$, and up to 5 for cluster size of $b = 5$. The original terrain has 59049 nodes, and 234740 edges.

2.4 Experiments on Terrains

While HPA* implicitly assumes grid-maps, the technique can be applied to terrains. The edge weights in the terrain are not used to identify way-points, but the intra-edges between way-points are constructed using the edge weights. Thus a way-point may be a poor choice, from the point of view of the cost of edges into it, but the intra-edges created for the next higher level will reflect the actual costs represented in the weighted graph.

In a general terrain, there may be no impassible obstacles, and these are the kind of terrains we investigate here. Thus, this part of the investigation will look at the path quality that results from possibly poor choice of way-points. We also take a look at the computational costs involved with constructing larger hierarchies than the previous experiments.

We used the 6 Brodatz terrains, sampled down from 729×729 to 243×243. We also used the Freeciv terrains, which were also 243×243. As stated above, this was done to keep runtimes reasonable; the implementation had difficulty constructing hierarchies of more than 2 levels for large terrains. We looked at two settings for the initial cluster size: $b = 5$ and $b = 10$; and unlike the previous experiments, we built hierarchies of height from 1 to 5 for $b = 5$ clusters, and from 1 to 4 for $b = 10$ clusters. The two initial cluster sizes were chosen to examine how way-points chosen affect the path quality. In a terrain, a cluster size of $b = 5$ puts way-points at the midpoint of each level 0 cluster boundary; and with $b = 10$, the way-points are at the corners of the level 0 cluster.

2.4.1 Pre-processing costs

The terrains are identical from the point of view of placing way-points, so the stuctures constucted by HPA* have a lot of similarity. For example, for all 9 terrains, the first level constucted for the $b = 10$ cluster-size will be the same; the higher levels will be identical as well, for as many levels as are constructed. For this reason, we can look at two structures, corresponding to the largest hierarchy for each initial cluster-size.

Table 7 show the data reported by HPA* for the terrains. We show the number of nodes at each level, and the number of inter- and intra-edges at each level. The table indicates, for example, that a 3 level hierarchy constructed using a $b = 10$ cluster size, has 1032 nodes at its top level, 2064 nodes at level 2, etc.

We observe that the 5 level hierarchy constructed using $b = 5$ clusters has roughly one-third the number of nodes in the original graph, and has roughly the number of edges in the original terrain. We also observe that the number of nodes decreases by roughly a factor of 2 at each level after the first, but the number of intra-edges increases from levels 1–3, with a decrease thereafter; the change in the number of intra-edges is consistent with the theoretical prediction, taking into account the fact that the dimension of the original

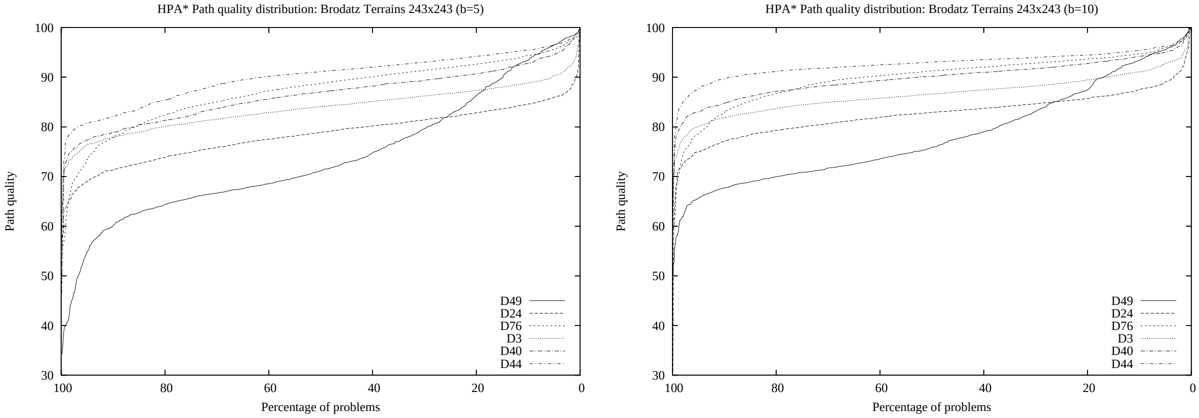

Figure 6: Cumulative distribution of path quality from applying HPA* to Brodatz terrains, using $b = 5$ clusters (left), and $b = 10$ (right).

terrain is not a multiple of the dimension of the cluster size.

2.4.2 On-line search performance

To investigate the path quality of HPA*, we ran 1000 pairs of random path queries in the HPA* hierarchy constructed for the Brodatz terrains, and the Freeciv terrains. The cumulative distribution of path quality is shown graphically in Figures 6 and 7. Table 8 provides a numerical perspective. Because of the way the hierarchy is constructed, and the way paths are refined, there is no difference in path quality when the height of the hierarchy is varied; the height only affects the search costs.

We observe that HPA* achieves path quality of 75% on at least 90% of the paths in 4 of the 6 Brodatz terrains when $b = 5$ clusters are used, and likewise in 5 of the 6 when $b = 10$ clusters are used. Path quality of 90% or better is achieved in at least 50% of the paths in one Brodatz terrain when the cluster size is $b = 5$, and in 3 Brodatz terrains when the cluster size is is $b = 10$. Path quality above 90% is rare in the Brodatz terrains. In the Freeciv terrains, we observe that path quality above 75% is rare.

We also observe that the path quality is generally better when a $b = 10$ cluster size is used, as compared to the $b = 5$ cluster size. This is a little counter-intuitive. One possible explanation is that the larger cluster size gives shorter paths better quality, because more of the path is exact, because more of it lies within the clusters where the endpoints are inserted. If this were true, we would expect a higher path quality for short paths. However, we do not observe this in the data. Instead, we observe that for longer paths, the path quality has much less variance for $b = 10$ than for $b = 5$. An alternative explanation is that the larger cluster size allows intra-edges to avoid more of high cost regions that may lie within a cluster, and therefore the cost of the intra-edges more closely reflect the cost of going through a cluster. In other words, smaller cluster sizes give the abstraction less flexibility.

2.5 On-line search costs

To demonstrate the the search costs of HPA*, we use the number of nodes expanded, as above for grid-maps. Because we are interested in the effect of the use of different numbers of levels for the hierarchy, the data

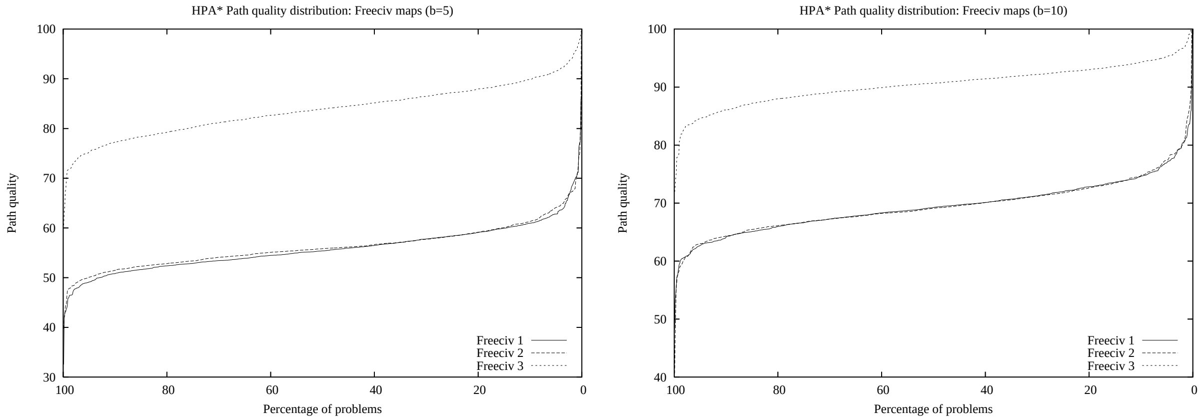

Figure 7: Cumulative distribution of path quality from applying HPA* to the Freeciv terrains, using $b = 5$ clusters (left), and $b = 10$ (right).

	$b = 5$					$b = 10$			
Map	99%	95%	90%	75%		99%	95%	90%	75%
D49	0.5	7.6	14.9	39.7		0.8	6.5	17.0	53.6
D24	0.2	0.3	0.8	74.6		0.4	0.9	3.4	94.9
D76	0.2	7.3	40.7	93.8		0.6	8.1	63.2	97.8
D3	0.2	0.8	5.4	96.5		0.5	1.5	16.7	99.1
D40	0.5	4.2	25.5	98.2		0.7	5.1	52.9	99.7
D44	0.7	13.6	61.5	99.3		0.7	13.5	88.8	99.8
Freeciv 1	0.1	0.1	0.1	0.7		0.2	0.2	0.3	8.7
Freeciv 2	0.2	0.2	0.2	0.5		0.3	0.3	0.4	9.2
Freeciv 3	0.3	1.6	9.6	95.4		0.9	5.7	59.2	99.7
Freeciv 4	0.3	1.5	5.8	87.8		0.7	4.0	33.0	99.7

Table 8: Cumulative distribution of path quality from applying HPA* to the Brodatz terrains, and the Freeciv terrains. On the left, the results of using an initial cluster size of $b = 5$, and $b = 10$ on the right.

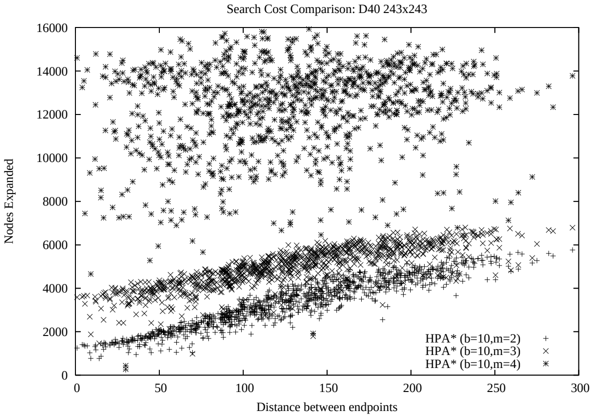

Figure 8: Comparison distribution of search costs from applying HPA* to Brodatz terrain D40, using $b = 10$ clusters. Only data using levels 2, 3, and 4 are shown.

are somewhat more complex to present, and we show the data in logical parts. We will start (for reasons that will become clear) by presenting only the search costs for the hierarchy above level 1.

Figure 8 gives the plot for one of the Brodatz terrains, for a cluster size of $b = 10$, showing the number of nodes expanded for the 1000 trials as a function of the distance between the end-points. We observe three almost distinct bands of points, which imply that the number of nodes expanded by search generally increases as the number of levels increases. This result is typical of the results from HPA* and appears also in the data obtained for the experiments using a cluster size of $b = 5$; see Figure 9. We also observe that the general trends in the three bands of points seems to be flatter as the number of levels in the hierarchy increases.

From these observations, we conclude that a significant cost in searching these hierarchies is the cost of inserting the endpoints into them. The insertion costs are high because of the number of nodes per cluster in the high levels of the hierarchy. Figure 9 shows the analagous for the case of $b = 5$ hierarchies.

Figures 10 and 11 show the search cost data for the same terrain, but using short hierarchies. Figure 10, which is the data for the $b = 10$ hierarchies, shows that for short paths, starting the search at level 1 is slightly less expensive than starting at level 2, but that for longer paths, it is less expensive to search starting at level 2. The situation is slightly more complicated for Figure 11, which shows the data for the $b = 5$ hierarchies. Search starting at level 1 is much more expensive than levels 2 or 3, for longer paths. For very short paths, starting search at level 3 is slightly more expensive than starting at level 1 or 2, but is less expensive for larger paths. This trend in the data is very similar across all the Brodatz terrains and the Freeciv terrains.

Figure 12 compares the cost of using A* to the cost of using the highest hierarchy (either 4 or 5 levels, depending on the cluster size), to the costs of using a shorter hierarchy, and A*. As the plot demonstrates, all of the HPA* hierarchies expand substantially fewer nodes that A*, but demonstrates that building a taller hierarchy is not always cost effective.

Finally, Table 9 gives a quantitative comparison of the search costs. The table is a cumulative distribution of the number of node expansions, expressed as a percentage of the number expansions of using A*. For both initial cluster sizes, more than 90% of all trials expanded as many as 50% of the number of nodes A* expanded, and none of the trials expanded less than 5% of the nodes expanded by A*. From the table, we can see that the combination of $b = 5$ and $m = 3$ results in lower search costs: just more than half of the

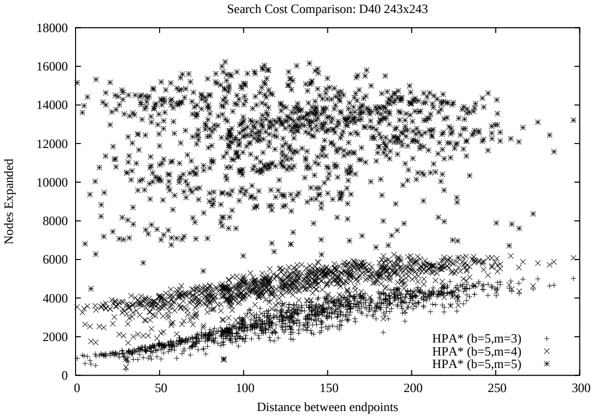

Figure 9: Comparison of search costs from applying HPA* to Brodatz terrain D40, using $b = 5$ clusters. Only data using levels 3, 4, and 5 are shown.

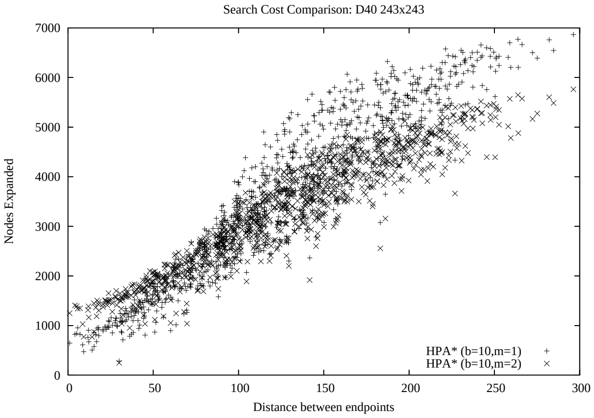

Figure 10: Comparison of search costs from applying HPA* to Brodatz terrain D40, using a $b = 10$ clusters (left). Only data using levels 1 and 2 are shown.

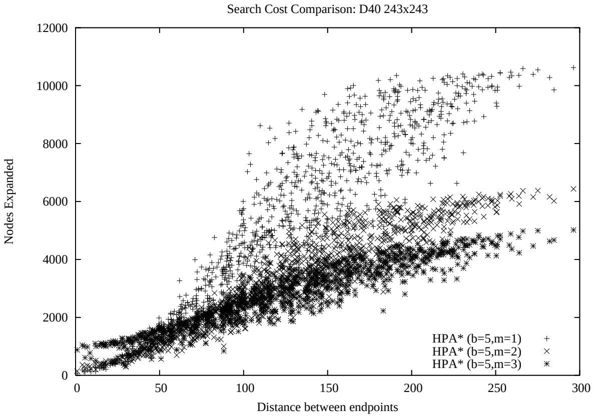

Figure 11: Comparison of search costs from applying HPA* to Brodatz terrain D40, using $b = 5$ clusters. Only data using levels 1, 2, and 3 are shown.

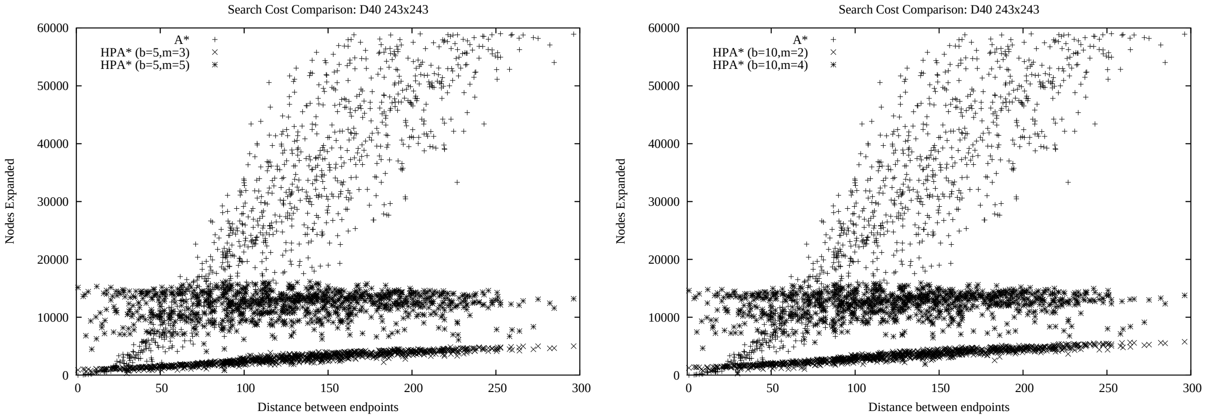

Figure 12: Comparison of search costs from applying HPA* to Brodatz terrain D40, using $b = 5$ clusters (left) and $b = 10$ clusters (right).

	$b = 5, m = 3$					$b = 10, m = 2$			
Map	5%	10%	25%	50%		5%	10%	25%	50%
D49	0	54.2	91.1	95.8		0	31.7	88.5	95.2
D24	0	56.7	92.1	96.6		0	38.6	88.6	94.7
D76	0	56.7	91.3	96.7		0	41.2	87.2	94.8
D3	0	53.9	91.8	96.3		0	35.1	88.2	94.7
D40	0	54.9	91.9	96.8		0	34.8	88.1	95.1
D44	0	52.5	91.6	96.3		0	33.5	87.8	95.3
Freeciv 1	0	59.1	91.8	96.4		0	39.7	88.0	94.9
Freeciv 2	0	58.9	91.6	96.3		0	38.9	88.0	95.0
Freeciv 3	0	56.2	92.7	96.4		0	36.2	88.2	95.5
Freeciv 4	0	55.2	92.1	95.9		0	36.1	88.1	95.1

Table 9: Cumulative distribution of search costs from applying HPA* to the Brodatz terrains, and the Freeciv terrains. The costs (columns) are expressed as a percentage of the cost of using A*.

trials needed to expand 10% of the number of nodes that A* expanded, whereas for $b = 10$ and $m = 2$, only about a third of the trials were so inexpensive.

3 Conclusions

In terms of path quality, the HPA* method finds paths of very high quality for a specific class of terrains we have been calling grid-maps. In the grid-maps we examined, including 4 mazes, 18 grid-maps derived by threshholding the Brodatz terrains and Freeciv grid-maps, a path quality of 95% or better is almost a certainty. This implies that the path smoothing operation makes up considerably for any inaccuracies that arise due to forcing the path through way-points. The search costs for the path-finding are very small compared to A*, on the order of 10 to 20 times fewer node expansions during search.

In terms of pre-processing costs, HPA* builds a hierarchy whose size, in terms of nodes, decreases by half with each level, but in terms of the number of edges, stays roughly constant except when the connectivity between adjacent clusters is highly constrained. We observed in the that the graphs in the upper levels of the hierarchy built from the Brodatz and Freeciv grid-maps do not decrease in size, since the dominant quantity is the number of intra-edges. We showed with a simple theoretical analysis of terrains without obstacles that this is exactly what should be expected. However, in the large maze example, where connectivity is more highly constrained, the number of intra-edges is on the same order of magnitude as the number of way-points. This is the main cost of pre-procecessing, since an intra-edge must be constructed for every pair of way-points in a cluster, and this involves search through the cluster.

The HPA* method implicitly assumes grid-maps as inputs, since edge weights are not considered in the selection of way-points. One possibility for application of HPA* to more general terrains is to approximate the terrain by removing edges above a given threshold, and setting the remaining edge weights to unit costs. The path quality in grid-maps constructed this way is very good, but when traced through the original terrain, the resulting path quality is not at all good; the path quality is better than 75% in less than 50% of the trials for half of the Brodatz grid-maps, and a path quality of 90% or better is rare in all Brodatz terrains, except D44. In other words, grid-maps are not a useful approximation to terrains, and terrain-specific path-finding techniques are needed.

We applied HPA* to terrains, by constructing a hierarchy as if the terrain had no obstacles, and using the edge weights to obtain the costs of the intra-edges between the default positions of the way-points. We applied this technique to the 6 Brodatz terrains, and the 4 Freeciv terrains, using two different initial cluster sizes, and a variety of hierarchy levels.

The path quality was substantially lower in terrains than in grid-maps. In two of the three Freeciv terrains, a path quality of 75% or better was rare. For the initial cluster size of $b = 5$, a path quality of 90% or better occurred in less than 50% of the trials for 5 of the 6 Brodatz terrains. The path quality for the initial-cluster size of $b = 10$ was better, with three of the six terrains achieving a path quality of 90% or better more than half the time. The path quality is below that of grid-maps primarily because the paths were forced to pass through arbitrarily chosen way-points, and the cost of doing so was substantially higher than in grid-maps. Where as the smoothing operation for grid-maps would improve the quality of a path, no such smoothing operation was used in our trials on terrains (and the one suggested by the authors of the method clearly does not apply to terrains).

Search costs for path-finding in terrains using HPA* are substantially lower that for A*, as for grid-maps. The initial clustering into $b = 5$ or $b = 10$ clusters dramatically reduced search costs for short paths, but the costs increased more than linearly with path length at this level. Additional levels help to reduce the search costs for long paths, but incur an over-head for short paths. We found that for the terrains we studied, constucting 2 levels above the original terrain was a good trade-off for an initial cluster size of $b = 10$, and 3 levels was a good trade-off for $b = 5$.

We found that there is a point after which adding levels to the hierarchy increases search costs (though costs never seem to approach those of A*). While the number of nodes per level decreases by a factor of 2 every level, the number of nodes per cluster increases by the same factor, and when there are enough nodes in a cluster, the cost of inserting the start- and end-point of a given path dominates the cost of search, by a very large margin. We emphasize that the height of the hierarchy incurs the higher over-head, because it is a function of cluster size, and not the size of the original terrain. The fact that the higher levels of the hierarchy do not decrease in size further supports the conclusion that only a few levels should be used.

In summary, HPA* achieves very high path quality in grid-maps, with acceptable pre-processing costs, and search costs. In terrains, the path quality achieved by HPA* is much lower. Search costs and pre-processing costs show that short hierarchies are preferred for HPA*.