Local Search Heuristics for Hop-constrained Directed Steiner Tree Problem†

Oleg Burdakov[1], Jonas Kvarnström[2], and Patrick Doherty[2]

[1] Department of Mathematics, Linköping University, Sweden,
oleg.burdakov@liu.se
[2] Department of Computer and Information Science, Linköping University, Sweden,
{jonas.kvarnstrom, patrick.doherty}@liu.se

Abstract. We consider the directed Steiner tree problem (DSTP) with a constraint on the total number of arcs (hops) in the tree. This problem is known to be NP-hard. Only heuristics can be applied in the case of its instances whose size is beyond the capacity of the existing exact algorithms. In our recent paper we used optimality conditions for developing an approach aimed at approximately solving hop-constrained DSTP. It can also be used for improving approximate solutions produced by other heuristic algorithms. Here we present specific label-correcting-type algorithms based on this approach and report preliminary results of their performance on a set of test problems. The test instances originate from 3D placement of unmanned aerial vehicles used for multi-target surveillance. They are characterized by a relatively small number of terminal nodes and a very large number of nodes and a huge number of arcs (above 10^8).

Keywords: Steiner trees, hop constraints, local search heuristics, label correcting algorithms

†The major part of this paper is composed of the text from [1].

Introduction

We consider the directed Steiner tree problem (DSTP) formulated as follows. Given a weighted directed graph with a selected root node and a set of terminal nodes, find a directed tree of minimal weight (or cost) which is rooted in the root node and spanning all terminal nodes as leaves. By directed tree, we mean here an arborescence, i.e. a directed graph in which there exists a unique directed path from the root node to any other node in the tree.

DSTP is known to be NP-hard. This imposes limitation on the size of problems that can be solved by the existing exact algorithms like those considered in [2–7]. Approximation algorithms are used in such cases (see [8–13]). Some strategies for improving approximate solutions are available in the literature (see, e.g., [14–17]).

In this paper, we will focus on the DSTP with a limitation on the total number of arcs (hops) by a given integer. We call it the *directed Steiner problem with hop constraints* (SPH). It is obviously more difficult than DSTP. SPH and similar problems were considered earlier in many publications, e.g. in [18] where the number of hops is limited in each directed path in the Steiner tree.

In [1], we presented new local search strategies aimed at improving SPH solutions produced by approximation algorithms. The strategies are based on optimality conditions. This local search is a further development of our approach to solving SPH appeared in [19]. It uses, like in [19], our label correcting algorithms proposed in [20–22] for finding Pareto optimal paths in the bi-criteria problem of minimizing the path cost and the number of hops.

Our work is motivated by the necessity of solving the problem of 3D placement of unmanned aerial vehicles (UAVs) used for multi-target monitoring and surveillance [19–22]. Our algorithms are able to efficiently exploit some features of the resulting large scale SPH problems, in which the tree cost is associated with the quality of the UAV placement, and the number of hops is directly related to the number of involved UAVs.

In this paper, we consider a few specific local search strategies based on the approach introduced in [1]. We study their behavior on a set of large SPH test instances related to the UAVs placement.

1 Problem Formulation and Optimality Conditions

Let $G = (N, A)$ be a weighted directed graph, where N is the set of nodes and A is the set of arcs. We suppose given: nonnegative costs c_{ij} associated with each arc $(i, j) \in A$, a root node $r \in N$ and a set of terminal nodes $T \subset N$. Let $S(\rho, \Lambda)$ stand for all directed trees in G whose root node is $\rho \in N$, and whose set of leaves is $\Lambda \subset N$. The trees in $S(r, T)$ are called *directed Steiner trees*. We shall use the abbreviated notation S for the set of all directed Steiner trees $S(r, T)$. Since the terminal nodes in T are required to be leaves of directed Steiner trees, we can assume without loss of generality that the outdegree of these nodes is zero. A collection of disjoint directed trees will be called a *forest*. For a subgraph

s of G, we denote the sum of the costs of all arcs in s and the total number of arcs (hops) in s by $c(s)$ and $h(s)$, respectively.

Given a positive integer H, the directed Steiner tree problem with hop constraints is to solve

$$\min\{c(s):\ s \in S,\ h(s) \leq H\}. \tag{1}$$

A subgraph of a directed tree s will be called a *subtree* of s if it is a directed tree, and for each of its nodes, the arcs in s that are outgoing for this node are all either internal or external with respect to the subgraph.

Consider any subtree $\bar{\sigma}$ of a tree $s \in S$. Let $\Lambda(\bar{\sigma})$ denote the set of the leaves of $\bar{\sigma}$. Consider also the graph denoted by $s \setminus \bar{\sigma}$ that is obtained from s by removing all the internal arcs and nodes of $\bar{\sigma}$, except the nodes $\Lambda(\bar{\sigma})$ and the root node of $\bar{\sigma}$. Observe that this removal splits s into two subgraphs, namely, a lower subtree $L(s \setminus \bar{\sigma})$ rooted in r and a forest rooted in $\Lambda(\bar{\sigma})$. Based on this observation, the key necessary optimality conditions of SPH that will be exploited in our local search strategies can be formulated as follows.

Theorem 1 ([1]). *Let s^* be an optimal solution to problem (1). Then any subtree $\bar{\sigma}$ of the Steiner tree s^* solves the problem*

$$\min_{\rho \in L(s^* \setminus \bar{\sigma})} \ \min_{\sigma \in S(\rho, \Lambda(\bar{\sigma}))} \{c(\sigma):\ h(\sigma) \leq h(\bar{\sigma})\}. \tag{2}$$

Given a Steiner tree $s \in S$ which approximately solves the SPH problem (1), Theorem 1 suggests that this approximate solution may be improved by choosing a proper subtree $\bar{\sigma}$ and then solving problem (2) for a better subtree, and hence a better Steiner tree. One can then continue by repeatedly choosing a new subtree and applying the same procedure.

In general, problem (2) may be of the same complexity as problem (1), unless the choice of the subtree $\bar{\sigma}$ is restricted by a certain class. For instance, if $\bar{\sigma}$ is a path, then (2) is reduced to finding for each $\rho \in L(s \setminus \bar{\sigma})$ a hop-constrained shortest path from ρ to the leaf of $\bar{\sigma}$. Each of these hop-constrained shortest path problems is polynomially solvable, e.g. with the use of modified Bellman-Ford algorithms considered in [23–25]. Thus, a natural requirement for the choice of $\bar{\sigma}$ is that the resulting problem (2) is to be efficiently solved. This was the case with the local search proposed in [19] for SPH, and this is the case with its improved version introduced in [1] and presented in the next section.

We shall use the following terms and notations. Node i in $s \in S$ is called a *Steiner node* of s if $i \notin \{r\} \cup T$. A Steiner node whose outdegree in s is greater than one is called a *star node*. The set of all star nodes in s is denoted by $N^\star(s)$. The nodes in the set $K(s) = N^\star(s) \cup \{r\} \cup T$ are called *key nodes*. A directed path in s is called a *key path* if it begins and ends in $K(s)$, and none of its internal nodes is a key node. For every star node i, there exist in s only one key path that ends in i and at least two key paths that begin in i. The subtree composed of these key paths is called the *star subtree* associated with $i \in N^\star(s)$, and it is denoted by s_i. Let $S^\star(\rho, \Lambda)$ stand for all star trees in G whose root node is $\rho \in N$ and the set of leaves is $\Lambda \subset N$.

Given $s \in S$, consider a directed tree whose nodes are the key nodes $K(s)$. The set of its arcs $A(s)$ is composed of the pairs (i, j) such that there exists a key path in s from $i \in K(s)$ to $j \in K(s)$. We call $R(s) = (K(s), A(s))$ a *reduced tree*. Obviously, r is the root of $R(s)$, and the set of its leaves is the same as in s, namely, T.

We shall use the following notations

$$i_- = \{j \in N : \ (j, i) \in A\} \quad \text{and} \quad i_+ = \{j \in N : \ (i, j) \in A\}$$

for the sets of immediate predecessors and successors of node $i \in N$ in the graph G, respectively.

Let i_-^R and i_+^R denote, respectively, the immediate predecessor and the set of all immediate successors of node i in the reduced tree R. Obviously, i_-^R is the starting node of the key path that ends in i, and i_+^R is the set of end nodes of the key paths that start in i.

We present below a weaker version of Theorem 1 for the reason that it suits better for motivating our local search. To clarify the relation between the next result and Theorem 1, we note that, for any star node i in s, we have $i_+^R = \Lambda(s_i)$.

Theorem 2 ([1]). *Let s^* be an optimal solution to problem (1). Then for each of its star nodes $i \in N^\star(s^*)$, the corresponding star subtree s_i^* of the Steiner tree s^* solves the problem*

$$\min_{\rho \in L(s^* \setminus s_i^*)} \ \min_{\sigma \in S^\star(\rho, i_+^R)} \{c(\sigma) : \ h(\sigma) \leq h(s_i^*)\}. \tag{3}$$

It should be emphasized that the optimality conditions presented by this theorem are necessary, but not sufficient conditions. This means that if a Steiner tree is such that each of its star subtrees is optimal, it is not, in general, an optimal solution to problem (1).

In the next section, we present auxiliary algorithms that will be involved in solving problems of the form (3).

2 Label Correcting Algorithms for Constrained Shortest Paths

In this section we consider directed paths in G from nodes in a given subset $L \subset N$ to a node $j \in N$. Let $P(L, j)$ denote the set of these paths. Consider the bi-criteria problem

$$\min\{(c(\pi), h(\pi)) : \ \pi \in P(L, j)\}. \tag{4}$$

Here, it is required to find directed paths $\pi \in P(L, j)$ that minimize, in the following sense, both the path cost $c(\pi)$ and the number of hops $h(\pi)$. We say that path π' *dominates* path π'' if the inequalities

$$c(\pi') < c(\pi'') \quad \text{and} \quad h(\pi') \leq h(\pi''),$$

or

$$c(\pi') \le c(\pi'') \quad \text{and} \quad h(\pi') < h(\pi'')$$

hold. By definition (see, e.g. [26, 27]), path $\pi^* \in P(L, j)$ is *Pareto optimal* if there is no other path $\pi \in P(L, j)$ which dominates π^*.

If $\pi^* \in P(L, j)$ is a Pareto optimal solution with $h(\pi^*) = k$, we call it a *k-hop Pareto path*. If there are multiple k-hop Pareto paths for a given k, it is sufficient for our purposes to finding only one of them. Let V_k denote the set of all nodes $j \in N$ for which there exists a k-hop Pareto path from L to j. If there exists a k-hop Pareto path $\pi^* \in P(L, j)$, we denote $g_k(j) = c(\pi^*)$. Given k, let k' be, if exists, the largest integer $k' < k$ for which there exists a k'-hop Pareto path from L to j, i.e. for which $j \in V_{k'}$. Denote

$$g_k^-(j) = \begin{cases} g_{k'}(j), \text{ if } k' \text{ exists,} \\ +\infty, \quad \text{otherwise.} \end{cases}$$

The next result presents optimality conditions for k-hop Pareto paths from the nodes in a set L to a node j. They are related to the availability of $(k-1)$-hop Pareto optimal paths from L to the predecessors of j. They are also related to the ability of those $(k-1)$-hop paths to provide j with a k-hop path which is shorter than all paths from L to j with less than k hops. The following optimality conditions were used in [1] for constructing and theoretically justifying our algorithms.

Theorem 3 ([1]). *A k-hop Pareto path $\pi^* \in P(L, j)$ exists iff the set $j_- \cap V_{k-1}$ is not empty and*

$$\min\{g_{k-1}(i) + c_{ij} : \ i \in j_- \cap V_{k-1}\} \ < \ g_k^-(j). \tag{5}$$

The predecessor of j in π^* is a minimizer for the left-hand-side of inequality (5), and $g_k(j)$ equals the value in the left-hand-side.

According to this theorem, if (5) holds for a node $j \in N$, then

$$g_k(j) = \min\{g_{k-1}(i) + c_{ij} : \ i \in j_- \cap V_{k-1}\}.$$

This equality can be viewed as a Bellman-type recurrence equation [24, 28–30].

The label correcting algorithms developed in [20–22] solve, in polynomial time, the bi-criteria problem (4) for sets L composed of only one node. We present here modifications of these algorithms intended to solve a more general problem in which L may contain arbitrarily many nodes, and in which there additionally are upper bounds $\bar{c}$ and $\bar{h}$ for path cost and and number of hops, respectively. To put it in another way, we are interested in finding Pareto optimal paths $\pi^* \in P(L, j)$ that solve the problem

$$\min\{(c(\pi), h(\pi)) : \ \pi \in P(L, j), \ c(\pi) \le \bar{c}, \ h(\pi) \le \bar{h}\}. \tag{6}$$

Theorem 3 can be easily extended to the optimal solutions to this problem.

Due to the presence of upper bounds in problem (6), some of the nodes in N may have an empty set of Pareto optimal paths. This allows for localizing the search over a certain subset of nodes, and hence decreasing the computational burden.

Each of the two algorithms introduced in [1] produce Pareto optimal solutions to problem (6) for the nodes $j \in N$ for which there exist feasible paths. The labels returned by the algorithms for node j represent a complete set of Pareto optimal paths from the nodes in L to node j. The labels have the form of triples

$$\{k,\ g_k(j),\ p_k(j)\}. \tag{7}$$

Here each of the triples corresponds to a k-hop Pareto path, with $g_k(j)$ and $p_k(j)$ standing for the path cost and the predecessor of j in this path, respectively. In our applications, the set of labels is very sparse. They typically contain just a few triples per node. For the theoretical background of the algorithms in this section, we refer to [20–22].

In the algorithms, the iteration number k corresponds to the number of hops, and $g(i)$ stands for the length of the currently shortest path from L to i of those paths so far considered in the solution process. The algorithms run at the k-th iteration over only those nodes i that belong to the set V_{k-1} and, whenever possible, improve the value of $g(j)$ for $j \in i_+$. If at that iteration the value of $g(j)$ changes at least once, i.e. when it becomes less than $g_k^-(j)$, this, by Theorem 3, indicates that there exists a k-hop Pareto path from L to j. The $g_k(j)$ and $p_k(j)$ are updated in accordance with the change of $g(j)$.

The first of the outlined algorithms can be formally presented as follows.

Algorithm 1.
$k \leftarrow 0$
for each $i \in N$ **do**
    if $i \in L$ **then**
        $g(i) \leftarrow 0,\ \ g_0(i) \leftarrow g(i),\ \ p_0(i) \leftarrow nil$
    else
        $g(i) \leftarrow +\infty,\ \ g_0(i) \leftarrow g(i)$
end (for)
$V_0 \leftarrow L$
for $k = 1, 2, \dots, \bar{h}$ **do**
    if $V_{k-1} = \emptyset$ **then** stop **else** $V_k \leftarrow \emptyset$
    for each $i \in V_{k-1}$ **do**
        for each $j \in i_+$ **do**
            $c \leftarrow g_{k-1}(i) + c_{ij}$
            if $c < g(j)$ and $c \le \bar{c}$ **then**
                $g(j) \leftarrow c,\ \ g_k(j) \leftarrow g(j),\ \ p_k(j) \leftarrow i,\ \ V_k \leftarrow V_k \cup \{j\}$
        end (for)
    end (for)
end (for)

The correctness of this algorithm and the next one can be justified by a direct analogy with the justification presented in [20, 21] for their prototypes.

The next algorithm takes advantage of using a solution to the following one-criterion shortest path problem

$$\min\{c(\pi):\ \pi \in P(L,j),\ c(\pi) \le \bar{c},\ h(\pi) \le \bar{h}\}. \tag{8}$$

It is obviously easier than the bi-criteria problem (6). There exist efficient algorithms for solving problem (8) (see e.g. [24, 29, 31]). Some of them, like Dijkstra's algorithm, are able to simultaneously solve this problem for all nodes j that have a feasible solution. They produce a forest of directed shortest paths rooted in L. Let K denote the height of the forest, i.e. the maximal number of hops over all directed paths that compose the forest. The set of nodes of the forest can be partitioned into subsets $N_0^*, N_1^*, \dots, N_K^*$, where each N_k^* consists of the nodes whose path from L in the forest has k hops.

Note that if node j belongs to N_k^*, then $g(j)$ does not change in Algorithm 1 after iteration k. In the next algorithm, the unnecessary check for improving $g(j)$ is avoided by disabling at the k-th iteration all incoming arcs for nodes $j \in N_k^*$.

Algorithm 2.
find a forest of shortest paths and set labels (7) for the forest nodes
$k \leftarrow 0$
for each $i \in N$ **do**
    if $i \in L$ **then**
        $g(i) \leftarrow 0$
    else
        $g(i) \leftarrow +\infty,\ \ g_0(i) \leftarrow g(i)$
end (for)
$V_0 \leftarrow L$
for $k = 1, 2, \dots, K-1$ **do**
    for each $j \in N_k^*$ **do**
        for each $i \in j_-$ **do**
            $i_+ \leftarrow i_+ \setminus \{j\}$
        end (for)
    end (for)
    $V_k \leftarrow N_k^*$
    for each $i \in V_{k-1}$ **do**
        for each $j \in i_+$ **do**
            $c \leftarrow g_{k-1}(i) + c_{ij}$
            if $c < g(j)$ and $c \le \bar{c}$ **then**
                $g(j) \leftarrow c,\ \ g_k(j) \leftarrow g(j),\ \ p_k(j) \leftarrow i,\ \ V_k \leftarrow V_k \cup \{j\}$
        end (for)
    end (for)
end (for)

It is easier to implement Algorithm 1, but the comparison of the prototypes of these two algorithms presented in [20, 21] was in favor of Algorithm 2.

3 Local Search

The local search strategies considered in [1] exploit the optimality conditions presented by Theorem 2. They are aimed at improving a given Steiner tree by improving its subtrees of the following two types. The first type refers to the star subtrees. The second one refers to the key paths that end in the terminal nodes. The latter will be called *terminal paths*.

3.1 Star Subtree Improvement

In [19], star subtrees were used as candidates for locally improving s. A candidate subtree is obtained in that paper by choosing a star node $i \in N^\star(s)$ and then solving the problem

$$\min_{\sigma \in S^\star(i_-^R, i_+^R)} \{c(\sigma) : \ h(\sigma) \le \bar{h}\}, \tag{9}$$

where $\bar{h}$ equals $h(s_i)$ or other proper value. As noted in [19], problem (9) can be solved in polynomial time with the use of the label correcting algorithm developed in [20–22] for finding Pareto optimal paths. It should be mentioned here that the local search suggested in [32] is also based on improving star subtrees, but it is not applicable to solving problem (9) because it is unable to handle hop constraints.

Consider some features of the local search based on solving problem (9). Let σ' solve problem (9). In practice, it is a rare case when the replacement of s_i in s by σ' results in ties. Assume that this is not the case for σ'. If the star node of σ' coincides with any of the key nodes in $N^\star(s) \setminus \{i\}$, then the total outdegree of the key nodes, denoted here by $d(s)$, decreases, otherwise it does not change. Therefore, if the initial Steiner tree $s \in S$ is such that the strict inequality $d(s) < d(s^*)$ holds, where s^* is an optimal Steiner tree, the same usually holds in practice for all subsequently generated improved approximations to the solution of the SPH problem (1). This means that the local search suggested in [32] has no chance to catch the topology of s^* characterized by $d(s^*)$, except the very rare case when ties appear. One can see that the new local search introduced in [1] and presented below is free of this shortcoming.

Our new local search is aimed, like in [19], at improving star subtrees s_i, but instead of solving problem (9), we suggest to solve problem (3), in which s^* is substituted by its current approximation s. This turns (3) into the problem

$$\min_{\rho \in L(s \setminus s_i)} \ \min_{\sigma \in S^\star(\rho, i_+^R)} \{c(\sigma) : \ h(\sigma) \le \bar{h}\}. \tag{10}$$

The choice $\bar{h} = h(s_i)$ in (10) is aimed at decreasing $c(s)$ and preventing from increasing $h(s)$. If to choose $\bar{h} > h(s_i)$, this, apart from a potentially more substantial decrease of $c(s)$, may result in an increase of $h(s)$ by up to $\bar{h} - h(s_i)$ hops. Any value of $\bar{h}$ below $h(s_i)$ admits a possible increase of $c(s)$ and serves to decrease $h(s)$. The last choice can be used for finding a Steiner tree feasible with respect to the hop constraint $h(s) \le H$.

Since the optimization is performed in (10) over all nodes $\rho \in L(s \setminus s_i)$ including the root of s_i, this wider choice of ρ offers a better possibility for improving s_i in comparison to solving problem (9). The local search based on (10) inherits the ability of the old local search to decrease $d(s)$, but it also allows for increasing it when neither the root of the solution to (10), nor its star node, coincide with any of the key nodes of s. The results of numerical experiments presented in [1] shows that the new local search is more efficient than the old one.

The new local search produces either an optimal solution to (10), which is the most typical case in practice, or even a better solution. This is because the set of the candidate solutions used by the local search is, in general, wider than $S^\star(\rho, i_+^R)$. Moreover, recall that in the optimality conditions presented by Theorem 2, it is assumed that the star tree is a subtree of an optimal Steiner tree, while the local search is applied to a Steiner tree, which is usually not optimal.

Consider the following idea of solving problem (10). It is based on labeling the nodes as described below. Each node in N gets, at most, $|i_+^R| + 1$ sets of Pareto optimal solutions in the form of labels (7). One set of labels results from solving problem (4) for $L = L(s \setminus s_i)$. For obtaining the other sets of labels, a reversed graph is used, i.e. the directed weighted graph $\hat{G} = (N, \hat{A})$ in which

$$\hat{A} = \{(i,j) : \ (j,i) \in A\} \quad \text{and} \quad \hat{c}_{ij} = c_{ji}.$$

The remaining sets of labels are produced by solving problem (4) in $\hat{G}$ for each leaf node $\tau \in i_+^R$, i.e. for $L = \{\tau\}$. Note that some of the sets of labels of node $j \in N$ may be empty if this node is infeasible in the corresponding problem (4). We say that node j is *completely labeled* if it has $|i_+^R| + 1$ nonempty sets of labels.

The required labels can be produced by running $|i_+^R| + 1$ times any of the two algorithms presented in the previous section. Assume as given a Steiner tree $s \in S$, one of its star nodes $i \in N^\star(s)$ and a parameter $\bar{h}$. Suppose that the optimal value of the objective function in (10) is bounded above by a given $\bar{c}$. If $\bar{h} = h(s_i)$, then in the mentioned algorithms it is natural to choose $\bar{c} = c(s_i)$, which would allow for narrowing the local search.

Consider a labeled node $j \in N$. Each of its labels corresponds, depending on the set containing this label, to a certain path either from $L(s \setminus s_i)$ to j, or from j to a leaf node of the star subtree s_i. Each label determines the cost $g_k(j)$ and the number of hops k of the corresponding path.

Suppose now that node j is completely labeled. Let Ξ stand for all possible combinations of $|i_+^R| + 1$ labels chosen from different sets of labels of this node. Consider a combination $\xi \in \Xi$. Each of the corresponding paths can be retrieved from the available labels (7) as described in [20, 21]. The paths compose a directed graph denoted by σ_ξ, which may not be a star tree. Moreover, σ_ξ does not necessarily belong to the set $S^\star(i_-^R, i_+^R)$, because σ_ξ may contain ties, the paths may have some arcs in common or some nodes of i_+^R may be internal for the paths.

Each combination ξ is characterized by the two numbers, denoted here by c_ξ and h_ξ, that are two sums of the g- and k-components of all labels in the combination ξ, i.e. they are, respectively, the total cost and number of hops of the corresponding paths. Because of the possible arcs in common, we have

$$c(\sigma_\xi) \le c_\xi \quad \text{and} \quad h(\sigma_\xi) \le h_\xi.$$

If σ_ξ has ties, this graph can be easily reduced, by cutting the ties, to a directed tree $\bar{\sigma}_\xi$ which has the same root node as σ_ξ and spans the nodes in i_+^R. This can be done, e.g., by finding in the graph σ_ξ a tree of directed shortest paths and then removing the arcs that do not belong to the paths from the root node to i_+^R. Since the arc costs are nonnegative, this results in the inequalities

$$c(\bar{\sigma}_\xi) \le c_\xi \quad \text{and} \quad h(\bar{\sigma}_\xi) < h_\xi.$$

For a completely labeled node j and the corresponding set of combinations Ξ, define

$$C(j) = \min\{c_\xi :\ \xi \in \Xi,\ h_\xi \le \bar{h}\}.$$

Thus, $C(j)$ stands for the total cost of the best combination of the paths for the given node j. Note that $C(j)$ may not exist even if node j is completely labeled. This occurs when j has no feasible combination of paths with respect to the total number of hops h_ξ.

Although the number of combinations in Ξ may grow exponentially with the problem size, the value of $C(j)$ can be computed in polynomial time. Consider the following procedure of computing $C(j)$ based on combining the available $|i_+^R|+1$ sets of labels. It starts with any set of labels, say the one that corresponds to $L(s \setminus s_i)$. Then the collection of the combined sets is successively extended by including one by one the sets from those corresponding to the nodes in i_+^R. At every step of the procedure, a combined set of labels of the $\{k, g\}$-type is constructed for the current collection of sets. The maximal number of such labels is $\bar{h}$. In each combined label $\{k, g\}$, the value of g equals the minimal total length of the paths from $L(s \setminus s_i)$ to j and from j to the currently collected nodes in i_+^R provided that the total number of hops of that paths is k. The combined labels are updated as follows.

Let

$$[\ldots, \{k', g'\}, \ldots]$$

be the current set of combined labels. It is initially composed of the labels that correspond to $L(s \setminus s_i)$. Let

$$[\ldots, \{k'', g''\}, \ldots]$$

be the set of labels that corresponds to the newly selected node in i_+^R. The new set of combined labels

$$[\ldots, \{k, g\}, \ldots]$$

is produced by summing pair-wise the labels $\{k', g'\}$ and $\{k'', g''\}$ with the aim to improve the label $\{k, g\}$, where $k = k' + k'' \le \bar{h}$. As a result,

$$g = \min\{g' + g'' :\ k' + k'' = k\}.$$

Note that there may not exist a label $\{k, g\}$ for some values of k. If the final set of combined labels is not empty, $C(j)$ is equal to the smallest value of g.

Consider a modification of the presented procedure that consists in maintaining, for every set of combined labels, a monotonic decrease of g with increase of k. This assumes removing the labels that violate the monotonicity, which speeds up the labeling process. One can easily verify that the modified procedure, as well as the original one, returns the correct value of $C(j)$.

Our new local search consists in labeling the nodes as described above, and then finding a node j^* which minimizes $C(j)$. The information about the predecessors contained in the labels allows for finding the collection of paths that provides the total path cost $C^* = C(j^*)$. In the next result, we compare the results of our local search with the solution to problem (10).

Lemma 1 ([1]). *Let σ^* be a solution to problem (10). Suppose that σ' is composed of the collection of paths that provides the minimal total path cost C^*. Then*

$$c(\sigma') \leq C^* \leq c(\sigma^*) \tag{11}$$

and $h(\sigma') \leq \bar{h}$. Moreover, if $|i_+^R| = 2$, then σ' solves problem (10).

3.2 Terminal Path Improvement

Suppose that the initial Steiner tree $s \in S$ is not of the star-type, which is the most typical case. Then s contains at least one star node such that all the key paths originating in this node are terminal paths. The process of improving star subtrees keeps these terminal nodes connected in the same way, i.e. via a joint star node, despite the fact that this may not be the case for the optimal Steiner tree. There is a chance to change this way of connecting the terminal nodes by trying to improve the terminal paths by solving the problem

$$\min_{\rho \in L(s \setminus s_t)} \min_{\pi \in P(\rho, t)} \{c(\pi) : \; h(\pi) \leq \bar{h}\}, \tag{12}$$

where $t \in T$, and s_t is the terminal path in s associated with t. The possible choices of $\bar{h}$ affects the values of $c(s)$ and $h(s)$ much like in the case of problem (10).

The local search that we suggest for improving terminal paths is closely related to our local search aimed at improving star subtrees. Namely, the nodes $j \in N$ are, first, suggested to be labeled with the triples (7) that originate from solving problem (4) in the revised graph $\hat{G}$ for $L = \{t\}$. Then a solution to problem (12) is obtained by choosing among the labeled nodes in $L(s \setminus s_t)$ the one with the smallest value of g_k, where $k \leq \bar{h}$. We shall refer to the two suggested local search strategies as *star*- and *path*-search.

Suppose that the star-search is applied to a star subtree which contains at least one key path. Then the labels required for solving problem (12) are readily available. This means that it looks reasonable to have this star-search followed by the path-search based on the available labels.

An alternative is to produce in advance the labels associated with the Pareto optimal paths from the root node r and to each of the terminal nodes T. This can be done before solving SPH (1), and then the stored labels can be used in the subsequent calculations.

4 Heuristic SPH Algorithms

Given a Steiner tree s, we shall refer to the star subtrees and key paths s_i, where $i \in N^\star(s) \cup T$, as *local components* of s.

We present here an approach to approximately solving the SPH problem (1). It consists in, first, finding an initial Steiner tree s, and then sequentially choosing a local component of s with the aim to improve it by the use of the local search introduced in the previous section.

The sequence of treating the local components of s assumes the availability of a queue of the corresponding key nodes. The queue is denoted here by Q, where $Q(1)$ stands for the node that defines the local component to be treated at the current iteration. The basic principle of including a key node i in the Q is that the corresponding local component s_i can potentially be improved by the local search. If the local search resulted in an improvement, the improved local component is replaced in s by the collection of paths that ensures the improvement. In the case of ties in s, they are cut to make it a Steiner tree. The changes in the Steiner tree during the current iteration imply the corresponding changes in the reduced tree. A few possible ways of arranging the order of nodes in the queue are discussed below. If Q is empty, this obviously means that none of the star subtrees or key paths of s can be improved by the local search.

The outlined heuristic approach is presented by the following algorithm.

Algorithm 3.
find an initial $s \in S$
compose a queue Q containing all nodes in $N^\star(s) \cup T$
while $Q \neq \emptyset$ **do**
  $i \leftarrow Q(1)$, delete $Q(1)$
  choose $\bar{h}$ for the local search
  try to improve s_i with the use of the local search
  if improvement **do**
    update s
    cut ties
    update $R(s)$
    update Q
  end (if)
end (while)

A large variety of heuristic SPH algorithms can be constructed on the base of this generic algorithm. Consider the following ways of its implementation.

Any fast DSTP heuristic, like the shortest path heuristic [33], can be used for generating an initial Steiner tree s. The resulting s may be infeasible, i.e.

the constraint $h(s) \leq H$ may be violated. With the aim of reducing $h(s)$, one can perturb all the arc costs c_{ij} by adding a positive scalar ε to each of them before using the chosen DSTP heuristic. This arc cost perturbation introduces implicitly a penalty which grows in proportion to the number of hops.

If the labels (7) associated with the Pareto optimal paths from the root node r and those terminating in T are produced in advance, their use can speed up the construction of the initial s. If these readily available labels are used in the shortest path heuristic, the increase of ε in the penalized path cost $g_k(j) + \varepsilon k$ could help to reduce $h(s)$, because this path cost is determined by the discussed above arc costs changed from c_{ij} to $c_{ij} + \varepsilon$. The underlying rationale is related to the property that the number of hops in the directed shortest path between any two given nodes calculated for the perturbed arc costs is monotonically decreasing or non-increasing with the increase of the ε (see [21]). Moreover, for all sufficiently large values of ε, the resulting shortest path has the minimal number of hops over all paths between that two nodes.

The parameter $\bar{h}$ of the local search plays an important role in Algorithm 3 in attaining and then maintaining the feasibility of s with respect to the hop constraint $h(s) \leq H$. The value of $\bar{h}$ changes from iteration to iteration, depending on $h(s)$ and $h(s_i)$, where s_i is the local component of s to be currently improved. In general,

$$\bar{h} = h(s_i) + \Delta h, \tag{13}$$

where the integer parameter Δh depends on $h(s)$.

If the initial or current Steiner tree s is infeasible, one can set for the local search $\Delta h = -1$ with the aim to decrease $h(s)$. Note that the local search may not find for the corresponding $\bar{h}$ any local solution with fewer hops than in s_i. Nevertheless, in this case, the available labels may allow the local search to find a local solution with a cost smaller than $c(s_i)$. It is natural to regard the Steiner tree as improved not only when $h(s)$ decreases, but also when $c(s)$ decreases even if $h(s)$ does not change. The sequence of iterations aimed at attaining the feasibility of s composes the first stage of the solution process.

As soon as the Steiner tree s becomes feasible, the solution process passes on to the second stage at which the feasibility of s is maintained. This can be done by choosing in (13) the parameter value

$$\Delta h = \begin{cases} 0, \text{ if } h(s) = H, \\ 1, \text{ if } h(s) < H. \end{cases} \tag{14}$$

At this stage, the Steiner tree is regarded as improved not only when $c(s)$ decreases, but also when $h(s)$ decreases even if $c(s)$ does not change.

Consider the case when $h(s) \leq H - 2$. Note that if a local component s_i cannot be improved for $\bar{h} = h(s_i) + 1$, it may still be improved for a larger value of $\bar{h}$. In view of this, it looks reasonable to continue the second stage when $h(s) \leq H - 2$ and there is no s_i that could be improved for $\bar{h} = h(s_i) + 1$. For the continuation, one can switch from (14) to the choice

$$\Delta h = \max\{H - h(s),\ 0\}.$$

Like (14), this choice maintains the feasibility of the Steiner tree.

According to Algorithm 3, if the updated s is not a tree, a directed Steiner tree must be obtained by properly cutting all ties in s. This can be done, for instance, by finding in the graph s a shortest path tree rooted in r and then removing the arcs in this tree that do not belong to the directed paths from r to the terminal nodes in T. Any cut of ties results in a further improvement of s in terms of both $c(s)$ and $h(s)$.

The determining constituents of the queue Q are a subset of key nodes that compose the Q and a linear order of these nodes. Before considering some ways of setting the order, consider the composition of the Q. By convention we will call a key node $i \in N^\star(s)$ *optimal* for a given $\bar{h}$, if it was proved by the local search that the corresponding local component s_i cannot be improved for this value of $\bar{h}$. Consider separately the two cases when $\Delta h \geq 0$ and when $\Delta h = -1$ at the current iteration. The Q is composed at this iteration of all nodes in $N^\star(s) \cup T$ except the key nodes i that are optimal for the $\bar{h}$ given by (13) in the former case, and for $\bar{h} = h(s_i)$ in the latter case.

Thus, it looks reasonable to store for each optimal key node the corresponding value or values of $\bar{h}$. Consider an iteration at which the local search improved the Steiner tree for a given $\bar{h}$. If there were no loops or arcs in common, and each of the produced Pareto optimal paths has at least one arc, the star node of the star tree composed by this paths is optimal for this $\bar{h}$. An optimal key node j may lose its optimality when the local search results in any change of s_j. In general, a key node j of the improved s is not optimal and must be placed in the queue, if j meets one of the following two requirements referring to the current iteration:

- at least one, but not all, of its key paths was produced by the local search;
- none of its key paths was produced by the local search, and $j^R_- \cup j^R_+$ changed.

This means that the root and terminal nodes of the local component that was improved by the local search are among the candidates for placing in the Q.

Before discussing possible ways of organizing the queue Q, we notice that the local search mostly does not result in changing the local components of s whose key nodes are not adjacent in $R(s)$ to the currently treated key node i. Each key node $i \in N^\star(s) \cup T$ is characterized by the number of hops $\nu(i)$ in the path from the root node r to the i in the reduced tree $R(s)$. One possibility is to keep the key nodes sorted in the Q in the increasing value of $\nu(i)$. In this case, the local components that are closer to the root node r are improved at the earlier iterations of Algorithm 3. At the later iterations, they are expected to be less affected by the changes in the local components that are more distant from the r. Similar expectations are related to the alternative way of organizing the queue, according to which the the key nodes are sorted in the Q in the decreasing value of $\nu(i)$.

Each key node $i \in N^\star(s) \cup T$ is characterized by the cost $c(s_i)$ and the number of hops $h(s_i)$. It would be natural to expect that the key node with the largest number of hops has a higher potential to improve the Steiner tree than the other key nodes in the queue. Similar expectations apply to the key node with the larges value of the associated cost. Thus, one more way of organizing the queue is related to sorting the nodes in the decreasing value of the associated number of hops or cost, or their combination.

If the labels (7) associated with T are available, it is relatively cheap to try to improve the terminal paths by checking these labels in the nodes of s. In view of this, if the recently improved local component contains a terminal path, the corresponding terminal node can be placed first in the Q. It is also reasonable to try to improve the terminal paths at the final stage of Algorithm 3. This can be ensured by keeping the terminal nodes in the end of the Q since the initiation of the queue. This admits that some of the terminal nodes may appear twice in the Q at the subsequent iterations.

5 Numerical Experiments

We tested our optimization algorithms on problems related to the 3D placement of UAVs used for multi-target monitoring and surveillance (see [1]). The test problems cover three different environments of the same size of 1000 times 1000 meters and a height of 80 meters. In the problem instance identifier they are referred to as world 10/11/12. They imitate in three different ways an urban environment with semirandom placement of 100 tall buildings. Our uniform discretization, the same for each environment, produced cells of four different sizes. They are referred to as grid 12/14/16/18, where the last is the finest grid. For each environment, a few combinations of base station position and 9 target positions were randomly generated, each combination is referred to as seed 0/1/.../11. The target positions were randomly generated in the way that they compose 3 clusters, each containing 3 targets with a cluster size of 100 meters. The arc cost imitates a combination of the signal degradation with the increasing distance, and also the closeness of UAVs to the obstacles (buildings). In computing the cost, we considered two values of the communication range, namely, 100 and 200 meters referred to as range 100/200.

In each problem instance, the value of the hop-limit H was chosen as the average of the numbers of hops of the two Steiner trees obtained by approximately solving DSTPs, in which $c(s)$ and $h(s)$ were separately minimized.

Our algorithms were implemented in C++. The numerical results presented here were produced on a standard PC with a 3.4 GHz Intel Core i7-3770 CPU and 32 GB of memory, of which not more 6 GB were actually used in our runs, and only one core was involved in the computational process.

We present here the results for five local search strategies referred to as hccost, hchops1, hchops2, hchops3 and hchops4. These strategies are based on the star-search only[3]. They have a common structure presented by the following stages.

1a) Find an initial Steiner tree $s \in S$.
1b) Find a feasible $s \in S$, i.e. such that $h(s) \leq H$.

[3] The path-search is not yet implemented.

2) While the queue Q is not empty, locally optimize $s \in S$ and keep s feasible.

In the strategies, the queue is organized as follows. Initially, all key nodes are marked as potentially not optimal, and they are sorted in the way that the closer they are to the root node in the reduced tree, the closer they are to the beginning of the Q. After a star-type subtree is improved, some of the key nodes become marked as potentially not optimal. Those of them which are not yet in the Q, are placed at its end, and they are sorted, like initially, depending on their closeness to the root.

The labeling in the strategies is performed by Algorithm 2. Depending on their preferences in choosing specific Pareto solutions, our strategies can be divided in the two groups: *cost-first* (hccost) and *hops-first* (hchops1/2/3/4).

In the both groups, a modified shortest path heuristic [33] is used on stage 1a). The difference is that, in the cost-first case, an initial s is obtained by approximately minimizing $c(s)$, while $h(s)$ is minimized in the hops-first case. The mentioned modification uses Pareto optimal alternatives obtained from the cost-hops labeling.

On stage 1b), the cost-first strategy hccost refines the initial s by using the star-search in which $c(s)$ is preferably minimized in the following sense. The current s is changed by the star-search either when it is able to maximally decrease $c(s)$, or, if impossible, when it is able to maximally decrease $h(s)$ without deteriorating $c(s)$. When s cannot be further improved in this way, the star-search continues with $\Delta h = -1$ in (13) until the feasibility is attained.

On stage 1b), the hops-first strategies refine the initial s using the star-search in which $h(s)$ is preferably minimized in the following sense. The current s is changed by the star-search either when it is able to decrease $h(s)$, or, if impossible, when it is able to maximally decrease $c(s)$ without deteriorating $h(s)$. In hchops1 and hchops3, the mentioned decrease is produced by the star-search with $\Delta h = -1$ in (13). In hchops2 and hchops4, if a decrease of $h(s)$ is possible, the maximal decrease of $h(s)$ is accepted. In hchops3 and hchops4, the process of decreasing $h(s)$ terminates as soon as the feasibility is attained. Strategies hchops1 and hchops2 do not stop in this case, and they continue while a decrease of $h(s)$ is possible.

On stage 2) of the five strategies, Δh is chosen for the star-search by formula (14) which allows for maintaining the feasibility.

The performance profiles of the five strategies are presented in Appendix A. One can see that hccost demonstrates the best performance from the view point of both the Steiner tree cost and the CPU time. The other strategies did not much differ between each other in their ability to minimize $c(s)$. Strategy hchops1 was the slowest. Fig. 2 shows that the CPU time of all these algorithms grows almost linearly with $|A|$. This property enables to approach solving practical UAV-related SPHs with billions of arcs.

Appendix B provides with more detailed results of running our algorithms on over 1400 problem instances.

References

1. Burdakov, O., Doherty, P., Kvarnström, J.: Local search for hop-constrained directed Steiner tree problem with application to UAV-based multi-target surveillance. In: Butenko, S., Pasiliao, E.L., Shylo, V. (eds.) Examining Robustness and Vulnerability of Networked Systems, pp. 26–50. IOS Press, (2014)
2. Wong, R.T.: A dual ascent approach for Steiner tree problems on a directed graph. Mathematical programming 28, 271-287 (1984)
3. Koch, T., Martin, A.: Solving Steiner tree problems in graphs to optimality. Networks 32, 207-232 (1998)
4. de Aragão, M.P., Uchoa, E., Werneck, R.F.: Dual heuristics on the exact solution of large Steiner problems. Electronic Notes in Discrete Mathematics 7, 150-153 (2001)
5. Polzin, T., Vahdati-Daneshmand, S.: Approaches to the Steiner Problem in Networks. In: Lerner, J., Wagner, D., Zweig, K.A. (eds.) Algorithmics of Large and Complex Networks. LNCS vol. 5515, pp. 81-103. Springer, Heidelberg (2009)
6. Gouveia, L., Simonetti, L., Uchoa, E.: Modeling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs. Mathematical Programming 128, 123-148 (2011)
7. Hougardy, S., Silvanus, J., Vygen, J.: Dijkstra meets Steiner: a fast exact goal-oriented Steiner tree algorithm. arXiv preprint arXiv:1406.0492 (2014).
8. D.-Z. Du, B. Lu, H. Ngo, P.M. Pardalos, Steiner tree problems, In: C.A. Floudas, P.M. Pardalos (eds.), *Encyclopedia of Optimization* **5**, Kluwer Academic Publishers, Dordrecht, 2001, 277–290.
9. D.-Z. Du, J.M. Smith, J.H. Rubinstein (eds), *Advances in Steiner Trees*, Kluwer Academic Publishers, Dordrecht, 2000.
10. E.N. Gordeev, O.G. Tarastsov, Steiner Problem: Survey. *Discr. Math. Appl.* **5** (1993), 339–364.
11. F. Hwang, D. Richards, P. Winter, *The Steiner Tree Problem*, North-Holland, Amsterdam, 1992.
12. A.O. Ivanov, A.A. Tuzhilin, *Minimal Networks: the Steiner Problem and its Generalizations*, CRC Press, Boca Raton FL, 1994.
13. C.A.S. Oliveira, P.M. Pardalos, A survey of combinatorial optimization problems in multicast routing. *Computers & Operations Research* **32** (2005), 1953–1981.
14. S.L. Martins, M.G.C. Resende, C.C. Ribeiro, P.M. Pardalos, A Parallel GRASP for the Steiner tree problem in graphs using a hybrid local search strategy, *J. of Global Optimization* **17** (2000), 267–283.
15. E. Uchoa, R.F. Werneck, Fast local search for Steiner trees in graphs, In: *Proc. of the Twelfth Workshop on Algorithm Engineering and Experiments* **10**, SIAM, Philadelphia, PA, 2010, 1–10.
16. M.G.A. Verhoeven, M.E.M. Severens, E.H.L. Aarts, Local search for Steiner trees in graphs, In: V.J. Rayward-Smith, et al. (eds), *Modern Heuristics Search Methods*, John Wiley and Sons, 1996, 117–129.
17. S. Voss, Modern heuristic search methods for the Steiner tree problem in graphs, In: D.-Z. Du, et al. (eds): *Advances in Steiner Trees*, Kluwer Academic Publishers, Dordrecht, 2000, 283–323.
18. S. Voss, The Steiner tree problem with hop constraints, *Annals of Operations Research* **86** (1999), 321–345.
19. P.-M. Olsson, J. Kvarnström, P. Doherty, O. Burdakov, K. Holmberg, Generating UAV communication networks for monitoring and surveillance, In: *Proc. of the*

11th Int. Conf. on Control, Automation, Robotics and Vision (ICARCV 2010), IEEE, 2010, 1070–1077.
20. O. Burdakov, P. Doherty, K. Holmberg, P.-M. Olsson, Optimal placement of UV-based communications relay nodes, *Journal of Global Optimization* **48** (2010), 511–531.
21. O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnström, P.-M. Olsson, Relay positioning for unmanned aerial vehicle surveillance, *International Journal of Robotics Research* **29** (2010), 1069–1087.
22. O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnström, P.-M. Olsson, Positioning unmanned aerial vehicles as communication relays for surveillance tasks, In: J. Trinkle, Y. Matsuoka, J.A. Castellanos (eds.), *Robotics, Science and Systems V*, MIT Press, 2010, 257–264.
23. E.L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, 1976.
24. R.K. Ahuja, T.L. Magnanti, J.B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice-Hall, 1993.
25. R. Guérin, A. Orda, Computing shortest paths for any number of hops, *IEEE/ACM Transactions on Networking* **10** (2002), 613–620.
26. K. Miettinen, *Nonlinear Multiobjective Optimization*, Springer, 1999.
27. M. Ehrgott, *Multicriteria Optimization*, Springer, Berlin, 2005.
28. R. Bellman, On routing problem, *Quarterly of Applied Mathematics* **16** (1958), 87–90.
29. D.P. Bertsekas, *Network Optimization: Continuous and Discrete Models*, Athena Scientific, 1998.
30. L.R. Ford, D.R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.
31. B.V. Cherkassky, A.V. Goldberg, T. Radzik, Shortest paths algorithms: Theory and experimental evaluation, *Mathematical Programming* **73** (1996), 129–174.
32. N.-P. Chen, New algorithms for Steiner tree on graphs, In: *Proc. IEEE Int. Symp. on Circuits and Systems*, IEEE, 1983, 1217–1219.
33. H. Takahashi, A. Matsuyama, An approximate solution for the Steiner problem in graphs, *Math. Japonica* **24** (1980), 573–577.

6 Appendix A: Performance characteristics

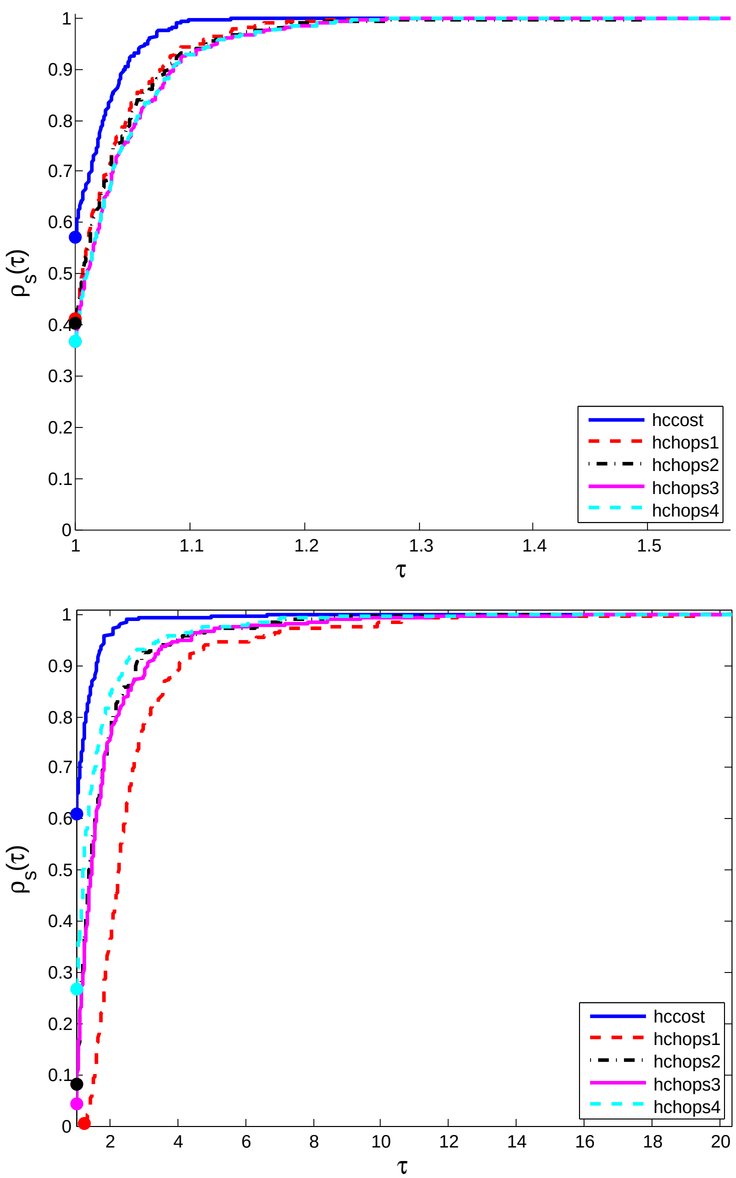

Fig. 1. Performance profiles for $c(s)$ (up) and CPU time (down) of the five strategies

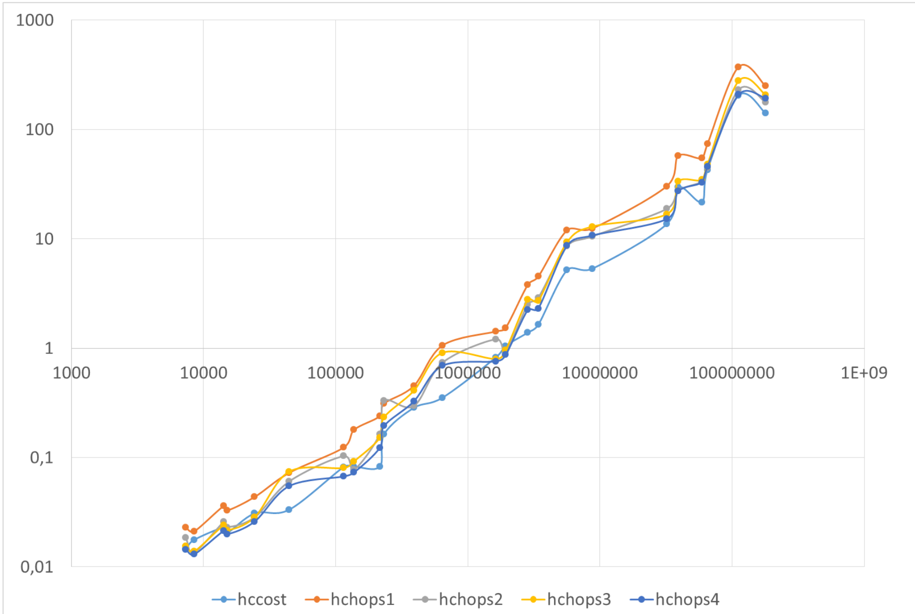

Fig. 2. Scalability of algorithms: log(time) as a function of log($|A|$). The values of $|A|$ are clustered, and each cluster is presented in the graph by its average value. The CPU time of each algorithm is averaged accordingly.

7 Appendix B: Test results

In the tables, we use the following notation:

- H is the hop-limit value,
- h_0 and c_0 are the number of hops and the cost of the initial tree produced by the modified shortest path heuristic,
- h and c are the final number of hops and the cost of the tree produced by the corresponding strategy,
- o is the number of successful attempts of the corresponding strategy to improve (optimize) star-subtree,
- Time is the total CPU time (in seconds) required for solving the problem (including the generation of the initial tree).

The tables, two for each combination of world = 10/11/12 and range = 100/200, contain big cells in which results for the five strategies hccost, hchops1, hchops2, hchops3 and hchops4 are presented. Each cell corresponds to a certain combination of Seed = 0/1/.../11 and grid = 12/14/16/18. Thus, each cell corresponds to a specific SPH instance. For example, in case of world=12, range=200, Seed=7 and grid=18, the instance identifier is
hc-relay-wo12-gr18-cr200-tr100-co1-ng9-se7.stp.

The presence of empty cells in Tables 5 and 7 are explained by the lack of connection between the root node and some of the terminal nodes, caused by the too crude grid.

Seed		grid=12, |N|=809				grid=14, |N|=3209				grid=16, |N|=12509				grid=18, |N|=55121			
		h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time
0		(|A|=14 238, H=35)				(|A|=215 314, H=31)				(|A|=2 841 323, H=31)				(|A|=59 069 617, H=29)			
	hccost	42/35	174528/171486	5	0.041	35/31	180867/167087	4	0.106	36/31	175679/164571	6	1.962	31/28	148892/139681	11	9.222
	hchops1	29/34	222910/182427	7	0.040	27/28	218729/180091	5	0.275	29/31	230247/171224	9	8.861	26/28	186511/139569	14	58.349
	hchops2	29/34	222910/182427	7	0.033	27/28	218729/180091	5	0.174	29/31	230247/179345	7	4.714	26/28	186511/139569	14	37.171
	hchops3	29/34	222910/182427	7	0.027	27/28	218729/180462	6	0.146	29/31	230247/179825	6	6.007	26/28	186511/139569	15	46.907
	hchops4	29/34	222910/182427	7	0.028	27/28	218729/180091	5	0.133	29/31	230247/179825	6	4.792	26/28	186511/139569	15	40.582
1		(|A|=14 253, H=32)				(|A|=215 344, H=29)				(|A|=2 841 418, H=28)				(|A|=59 070 181, H=29)			
	hccost	37/32	156727/158691	5	0.026	31/29	155044/151903	5	0.105	31/28	154618/145076	7	1.759	31/29	153406/146620	8	39.927
	hchops1	27/32	191059/151822	7	0.033	27/29	200736/157306	5	0.253	25/28	188236/146900	6	4.077	26/29	197611/149650	6	91.663
	hchops2	27/32	191059/151822	7	0.026	27/29	200736/156823	6	0.160	25/28	188236/146900	6	2.980	26/29	197611/149650	6	59.176
	hchops3	27/32	191059/151822	7	0.024	27/29	200736/157285	5	0.188	25/28	188236/146900	7	3.055	26/29	197611/151420	6	42.597
	hchops4	27/32	191059/151822	7	0.023	27/29	200736/157285	5	0.151	25/28	188236/146900	7	2.493	26/29	197611/151420	6	44.936
2		(|A|=14 218, H=29)				(|A|=215 209, H=28)				(|A|=2 840 897, H=28)				(|A|=59 068 157, H=27)			
	hccost	33/29	149027/149253	4	0.014	31/28	151052/146250	4	0.071	30/27	149897/140261	6	0.376	30/27	148876/139638	8	22.423
	hchops1	29/29	201343/148655	6	0.034	26/28	174749/147462	3	0.175	25/28	181293/144591	7	2.693	24/27	176733/140373	8	46.666
	hchops2	29/29	201343/139160	6	0.024	26/28	174749/147462	3	0.098	25/28	181293/144591	7	1.637	24/27	176733/140373	8	30.013
	hchops3	29/29	201343/139160	5	0.029	26/28	174749/147462	4	0.099	25/28	181293/144591	8	1.657	24/27	176733/140373	9	28.573
	hchops4	29/29	201343/139160	5	0.024	26/28	174749/147462	4	0.071	25/28	181293/144591	8	1.385	24/27	176733/140373	9	27.886
3		(|A|=14 256, H=30)				(|A|=215 352, H=27)				(|A|=2 841 468, H=26)				(|A|=59 070 259, H=26)			
	hccost	34/28	149467/138929	11	0.035	30/27	150228/138248	4	0.060	30/26	146066/135434	9	1.458	30/26	142821/129921	10	35.803
	hchops1	28/30	189701/144482	12	0.043	26/27	180530/135201	8	0.196	25/26	190578/139336	8	3.719	25/26	179448/128357	9	52.558
	hchops2	28/30	189701/144482	12	0.026	26/27	180530/135201	8	0.108	25/26	190578/139336	8	2.316	25/26	179448/128357	9	28.881
	hchops3	28/30	189701/145864	8	0.025	26/27	180530/135201	10	0.109	25/26	190578/138284	11	2.255	25/26	179448/127604	12	31.754
	hchops4	28/30	189701/145265	12	0.022	26/27	180530/135201	10	0.096	25/26	190578/138284	11	1.731	25/26	179448/127604	12	29.391
4		(|A|=14 234, H=28)				(|A|=215 283, H=26)				(|A|=2 841 160, H=26)				(|A|=59 069 542, H=22)			
	hccost	31/27	142349/135988	5	0.023	29/26	138038/132157	5	0.086	31/26	154717/139484	11	2.141	25/22	125800/113321	7	26.404
	hchops1	25/28	178122/135699	4	0.030	24/26	172946/136117	7	0.235	24/26	180325/136092	14	4.620	21/22	142959/107886	12	45.175
	hchops2	25/28	178122/135699	4	0.023	24/26	172946/136117	7	0.176	24/26	180325/136092	14	3.471	21/22	142959/107886	12	28.624
	hchops3	25/28	178122/135699	4	0.020	24/26	172946/136117	7	0.155	24/26	180325/136092	14	3.586	21/22	142959/107886	12	25.625
	hchops4	25/28	178122/135699	4	0.019	24/26	172946/136117	7	0.118	24/26	180325/136092	14	2.966	21/22	142959/107886	12	26.772
5		(|A|=14 230, H=33)				(|A|=215 228, H=30)				(|A|=2 840 980, H=31)				(|A|=59 068 531, H=30)			
	hccost	39/33	167761/164342	5	0.026	34/30	173357/163354	7	0.134	35/31	169068/160899	7	1.533	34/30	171397/159680	11	40.288
	hchops1	28/29	205971/190109	2	0.022	27/30	207463/157926	6	0.295	27/31	215279/170445	7	3.976	27/30	213352/156384	10	53.957
	hchops2	28/29	205971/190109	2	0.013	27/30	207463/157926	7	0.230	27/31	215279/170445	7	3.009	27/30	213352/158701	9	32.598
	hchops3	28/29	205971/190109	2	0.012	27/30	207463/157926	7	0.247	27/31	215279/170445	8	2.940	27/30	213352/157854	14	35.728
	hchops4	28/29	205971/190109	2	0.014	27/30	207463/157926	7	0.172	27/31	215279/170445	8	2.657	27/30	213352/157854	14	39.898

Table 1. Test Results: world=10, range=100, part 1

		grid=12, |N|=809				grid=14, |N|=3209				grid=16, |N|=12509				grid=18, |N|=55121			
Seed		h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time
6			(|A|=14 247, H=35)				(|A|=215 321, H=34)				(|A|=2 841 329, H=36)				(|A|=59 069 653, H=28)		
	hccost	46/35	190949/183569	6	0.032	38/34	198576/199568	6	0.152	39/36	196745/194939	8	1.865	31/28	151370/141049	4	7.797
	hchops1	31/35	229396/208480	6	0.039	31/34	239088/200603	4	0.278	32/36	250463/200996	8	3.489	25/28	183994/144196	7	51.980
	hchops2	31/35	229396/209784	5	0.035	31/34	239088/200603	4	0.190	32/36	250463/200996	8	2.715	25/28	183994/144196	7	37.533
	hchops3	31/35	229396/209784	6	0.025	31/34	239088/200603	5	0.236	32/36	250463/200996	9	2.957	25/28	183994/144196	8	40.911
	hchops4	31/35	229396/209784	6	0.024	31/34	239088/200603	5	0.175	32/36	250463/200996	9	2.314	25/28	183994/144196	8	36.776
7			(|A|=14 260, H=21)				(|A|=215 361, H=21)				(|A|=2 841 496, H=20)				(|A|=59 070 735, H=20)		
	hccost	23/21	105224/97554	3	0.005	23/21	106246/99204	3	0.022	23/20	105916/95645	5	0.578	22/20	100396/93555	7	5.266
	hchops1	21/21	133026/92862	4	0.024	21/21	120734/94710	10	0.187	19/20	116586/94246	7	2.068	19/20	125819/93451	7	34.499
	hchops2	21/21	133026/92862	4	0.021	21/21	120734/94710	10	0.140	19/20	116586/94246	7	1.164	19/20	125819/93451	7	14.588
	hchops3	21/21	133026/100061	4	0.013	21/21	120734/96924	8	0.047	19/20	116586/94246	6	1.002	19/20	125819/93451	9	16.684
	hchops4	21/21	133026/100061	4	0.013	21/21	120734/96924	8	0.047	19/20	116586/94246	6	0.891	19/20	125819/93451	9	15.853
8			(|A|=14 239, H=33)				(|A|=215 295, H=31)				(|A|=2 841 245, H=31)				(|A|=59 069 594, H=31)		
	hccost	42/33	175279/168273	5	0.025	35/31	176237/165209	8	0.095	36/31	181877/163041	9	1.472	36/31	181399/164577	11	31.980
	hchops1	33/33	242791/168273	11	0.073	29/31	217113/163895	7	0.338	28/31	223639/184203	6	4.474	28/31	219134/161690	5	62.043
	hchops2	33/33	242791/168273	9	0.041	29/31	217113/163895	7	0.258	28/31	223639/184203	6	4.228	28/31	219134/161690	5	41.140
	hchops3	33/33	242791/168273	8	0.051	29/31	217113/163895	6	0.166	28/31	223639/184203	7	4.938	28/31	219134/161690	6	44.538
	hchops4	33/33	242791/168273	8	0.037	29/31	217113/163895	7	0.159	28/31	223639/184203	7	3.776	28/31	219134/161690	6	38.478
9			(|A|=14 247, H=27)				(|A|=215 312, H=24)				(|A|=2 841 341, H=24)				(|A|=59 070 011, H=24)		
	hccost	29/27	125987/124580	2	0.012	27/24	124457/117782	4	0.086	27/24	126258/116588	6	0.184	27/24	124401/117137	8	20.832
	hchops1	23/27	157958/128881	7	0.029	25/24	157053/115774	10	0.210	22/24	159324/119480	9	1.829	22/24	148157/114595	8	46.033
	hchops2	23/27	157958/128881	7	0.021	25/24	157053/114855	10	0.113	22/24	159324/119480	9	1.299	22/24	148157/114595	8	31.034
	hchops3	23/27	157958/128881	7	0.024	25/24	157053/116605	8	0.123	22/24	159324/119480	9	1.445	22/24	148157/114595	9	32.022
	hchops4	23/27	157958/128881	7	0.015	25/24	157053/116605	8	0.092	22/24	159324/119480	9	1.115	22/24	148157/114595	9	28.704
10			(|A|=14 254, H=27)				(|A|=215 352, H=24)				(|A|=2 841 411, H=24)				(|A|=59 070 494, H=24)		
	hccost	29/27	123743/119363	2	0.004	26/24	124471/116631	9	0.044	26/24	124061/117309	7	0.952	26/24	126292/116779	13	12.784
	hchops1	26/26	165414/117864	5	0.027	22/24	142277/117571	9	0.159	22/24	147462/115725	8	2.701	22/24	149456/118441	10	40.994
	hchops2	26/26	165414/117864	5	0.017	22/24	142277/117571	9	0.098	22/24	147462/115725	8	1.046	22/24	149456/118441	10	17.225
	hchops3	26/26	165414/117864	6	0.014	22/24	142277/116631	10	0.133	22/24	147462/115725	9	1.466	22/24	149456/118441	11	21.865
	hchops4	26/26	165414/117864	6	0.014	22/24	142277/116631	10	0.088	22/24	147462/115725	9	1.136	22/24	149456/118441	11	18.852
11			(|A|=14 233, H=26)				(|A|=215 310, H=25)				(|A|=2 841 276, H=24)				(|A|=59 069 726, H=26)		
	hccost	31/25	135162/129851	7	0.032	27/25	136009/122039	4	0.027	28/24	136112/123437	11	2.381	28/26	133075/125330	5	6.636
	hchops1	22/26	157284/128377	6	0.040	23/25	157039/123414	11	0.264	23/24	165254/124014	6	3.053	23/25	159052/120604	9	70.504
	hchops2	22/26	157284/128377	6	0.028	23/25	157039/123414	11	0.221	23/24	165254/124014	6	1.473	23/25	159052/120604	9	44.506
	hchops3	22/26	157284/128377	5	0.026	23/25	157039/123414	12	0.163	23/24	165254/123766	7	2.001	23/25	159052/120604	10	47.720
	hchops4	22/26	157284/128377	4	0.025	23/25	157039/123414	12	0.170	23/24	165254/123766	7	1.606	23/25	159052/120604	10	46.828

Table 2. Test Results: world=10, range=100, part 2

Seed		grid=12, |N|=809				grid=14, |N|=3209				grid=16, |N|=12509				grid=18, |N|=55121			
		h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time
0		(|A|=44 494, H=25)				(|A|=642 774, H=24)				(|A|=8 738 480, H=24)				(|A|=178 747 003, H=22)			
	hccost	27/25	62887/62858	5	0.035	27/24	63108/61295	6	0.334	27/24	65133/64319	5	5.577	26/22	62569/57016	11	157.504
	hchops1	22/25	125264/61396	8	0.099	22/24	138838/60801	11	1.330	22/24	150725/60765	14	20.695	20/22	116860/57085	11	241.868
	hchops2	22/25	125264/61178	8	0.073	22/24	138838/60801	10	0.985	22/24	150725/60765	14	16.637	20/22	116860/57085	11	179.591
	hchops3	22/25	125264/63292	8	0.089	22/24	138838/60213	15	1.145	22/24	150725/63648	11	16.451	20/22	116860/57396	13	234.498
	hchops4	22/25	125264/63292	8	0.068	22/24	138838/60213	15	0.916	22/24	150725/63648	11	14.327	20/22	116860/57396	13	202.351
1		(|A|=44 509, H=23)				(|A|=642 804, H=22)				(|A|=8 738 575, H=21)				(|A|=178 747 567, H=22)			
	hccost	26/23	60579/63450	7	0.037	26/22	60168/57252	9	0.322	27/21	61890/57433	9	7.847	25/22	62805/57712	11	173.322
	hchops1	20/23	101628/62000	9	0.080	20/22	100643/57647	8	1.127	20/21	110101/60112	15	11.254	20/22	107277/60433	9	227.514
	hchops2	20/23	101628/62000	9	0.070	20/22	100643/57647	8	0.702	20/21	110101/60112	15	10.075	20/22	107277/60433	9	142.085
	hchops3	20/23	101628/62000	9	0.085	20/22	100643/56631	9	0.970	20/21	110101/57472	12	14.692	20/22	107277/60792	13	240.470
	hchops4	20/23	101628/62000	9	0.069	20/22	100643/56631	9	0.779	20/21	110101/57472	12	12.876	20/22	107277/60792	13	215.245
2		(|A|=44 474, H=22)				(|A|=642 669, H=21)				(|A|=8 738 054, H=21)				(|A|=178 745 543, H=22)			
	hccost	26/22	60884/61413	5	0.038	25/21	60570/59234	9	0.386	25/21	61623/57187	10	5.993	24/22	59697/58648	6	143.317
	hchops1	17/22	85281/61413	7	0.098	17/21	100795/57454	7	1.010	17/21	101186/57292	9	9.456	18/22	105482/63441	6	194.823
	hchops2	17/22	85281/61413	7	0.082	17/21	100795/57454	7	0.682	17/21	101186/57292	9	8.877	18/22	105482/63441	6	129.339
	hchops3	17/22	85281/61413	8	0.085	17/21	100795/57454	8	0.697	17/21	101186/57292	10	9.967	18/22	105482/63441	7	162.748
	hchops4	17/22	85281/61413	7	0.071	17/21	100795/57454	8	0.557	17/21	101186/57292	10	8.768	18/22	105482/63441	7	138.831
3		(|A|=44 512, H=22)				(|A|=642 812, H=22)				(|A|=8 738 625, H=21)				(|A|=178 747 645, H=20)			
	hccost	27/22	60546/57950	8	0.033	25/22	58139/52765	5	0.089	25/21	59711/53233	10	3.601	25/20	58885/53540	7	164.838
	hchops1	18/22	96078/57130	6	0.054	21/21	107962/53328	8	0.879	18/21	102820/53331	6	7.965	18/20	107072/53492	13	277.504
	hchops2	18/22	96078/57130	6	0.043	21/22	107962/52855	12	0.766	18/21	102820/53331	6	6.585	18/20	107072/53492	13	203.683
	hchops3	18/22	96078/57130	7	0.052	21/22	107962/52854	12	1.022	18/21	102820/53186	9	10.824	18/20	107072/53492	14	252.545
	hchops4	18/22	96078/57130	7	0.037	21/22	107962/52854	12	0.608	18/21	102820/53186	9	8.613	18/20	107072/53492	14	215.567
4		(|A|=44 490, H=23)				(|A|=642 743, H=21)				(|A|=8 738 317, H=19)				(|A|=178 746 928, H=20)			
	hccost	26/23	59279/55233	7	0.050	23/21	55193/54545	6	0.258	26/19	64941/53985	10	7.426	20/18	54703/48096	8	31.655
	hchops1	18/23	90300/56429	7	0.054	17/21	91213/51733	7	1.045	16/19	92146/53901	5	10.417	18/20	97536/49594	10	220.914
	hchops2	18/23	90300/56429	7	0.052	17/21	91213/51733	7	0.722	16/19	92146/53901	5	9.114	18/20	97536/49594	10	146.529
	hchops3	18/23	90300/56429	7	0.051	17/21	91213/51733	8	0.805	16/19	92146/53901	5	10.936	18/20	97536/49594	11	141.568
	hchops4	18/23	90300/56429	7	0.040	17/21	91213/51733	8	0.611	16/19	92146/53901	5	9.110	18/20	97536/49594	11	145.581
5		(|A|=44 486, H=24)				(|A|=642 688, H=23)				(|A|=8 738 137, H=23)				(|A|=178 745 917, H=22)			
	hccost	27/24	64607/63997	4	0.025	28/23	68282/64781	8	0.492	28/23	66975/60232	12	5.051	27/22	65738/61354	11	226.871
	hchops1	20/24	114091/62006	8	0.067	21/23	118923/60256	10	1.441	21/23	115279/60232	15	15.387	18/22	116426/61794	14	286.319
	hchops2	20/24	114091/62006	8	0.059	21/23	118923/60256	10	0.940	21/23	115279/60232	15	13.357	18/22	116426/61794	14	234.218
	hchops3	20/24	114091/62006	9	0.081	21/23	118923/59548	10	1.174	21/22	115279/67051	8	12.847	18/22	116426/61794	15	258.217
	hchops4	20/24	114091/62006	9	0.049	21/23	118923/60266	14	1.171	21/22	115279/67051	8	9.913	18/22	116426/61794	15	264.239

Table 3. Test Results: world=10, range=200, part 1

Seed		grid=12, |N|=809				grid=14, |N|=3209				grid=16, |N|=12509				grid=18, |N|=55121			
		h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time
6		(|A|=44 503, H=26)				(|A|=642 781, H=25)				(|A|=8 738 486, H=27)				(|A|=178 747 039, H=21)			
	hccost	29/26	65760/70561	4	0.046	31/25	70224/70252	10	0.651	31/27	70050/64335	13	10.564	27/21	61801/56560	11	228.731
	hchops1	23/26	139931/74371	8	0.102	22/25	131198/71364	8	1.474	23/27	145197/67605	14	16.716	18/21	108134/58323	6	277.838
	hchops2	23/26	139931/74371	6	0.083	22/25	131198/71364	7	1.131	23/27	145197/67605	14	17.185	18/21	108134/58323	6	194.158
	hchops3	23/26	139931/74371	6	0.095	22/25	131198/72018	8	1.014	23/27	145197/67605	15	18.853	18/21	108134/58323	7	195.440
	hchops4	23/26	139931/74371	5	0.067	22/25	131198/72018	8	0.786	23/27	145197/67605	15	16.883	18/21	108134/58323	6	171.683
7		(|A|=44 516, H=19)				(|A|=642 821, H=17)				(|A|=8 738 653, H=16)				(|A|=178 748 121, H=17)			
	hccost	22/19	52756/46307	5	0.009	19/17	48203/45326	8	0.227	19/16	49021/44903	14	2.809	19/17	50804/43909	13	41.458
	hchops1	17/19	62749/46543	9	0.045	16/17	62362/43347	5	0.496	16/16	75228/44995	12	8.497	15/17	74707/44480	8	165.085
	hchops2	17/19	62749/46543	9	0.033	16/17	62362/43347	5	0.303	16/16	75228/44995	12	4.716	15/17	74707/44480	8	74.724
	hchops3	17/19	62749/46543	9	0.037	16/17	62362/43414	7	0.342	16/16	75228/44499	5	6.232	15/17	74707/44480	9	75.825
	hchops4	17/19	62749/46543	9	0.030	16/17	62362/43414	7	0.298	16/16	75228/44499	5	4.696	15/17	74707/44480	9	74.630
8		(|A|=44 495, H=24)				(|A|=642 755, H=23)				(|A|=8 738 402, H=24)				(|A|=178 746 980, H=24)			
	hccost	29/24	64076/66492	6	0.044	28/23	61437/62683	9	0.676	29/24	65974/60759	10	7.933	29/24	66234/61440	8	204.957
	hchops1	22/24	123683/66410	7	0.083	22/23	134104/60303	9	1.072	20/24	136874/59244	7	14.303	20/24	135789/60106	13	388.718
	hchops2	22/24	123683/65771	9	0.075	22/23	134104/60303	9	0.735	20/24	136874/58548	5	8.130	20/24	135789/60106	13	299.523
	hchops3	22/24	123683/65771	13	0.153	22/23	134104/62738	12	0.892	20/24	136874/58548	6	10.834	20/24	135789/60106	14	289.510
	hchops4	22/24	123683/65771	13	0.099	22/23	134104/62738	12	0.826	20/24	136874/58548	6	8.513	20/24	135789/60106	14	287.662
9		(|A|=44 503, H=21)				(|A|=642 772, H=19)				(|A|=8 738 498, H=19)				(|A|=178 747 397, H=19)			
	hccost	24/21	55782/53185	5	0.019	23/19	53519/48553	7	0.087	22/19	53571/49217	5	2.537	22/19	54911/51681	9	95.422
	hchops1	18/21	99757/51737	10	0.053	17/19	84266/49991	7	0.931	19/19	96903/50722	11	7.228	18/19	88989/50457	11	217.266
	hchops2	18/21	99757/51737	10	0.041	17/19	84266/49991	7	0.574	19/19	96903/50722	11	4.154	18/19	88989/50457	11	117.099
	hchops3	18/19	99757/52692	8	0.038	17/19	84266/49991	8	0.818	19/19	96903/51951	11	11.230	18/19	88989/51681	13	153.102
	hchops4	18/19	99757/52692	8	0.033	17/19	84266/49991	8	0.509	19/19	96903/51951	11	9.158	18/19	88989/51681	13	139.492
10		(|A|=44 510, H=20)				(|A|=642 812, H=19)				(|A|=8 738 568, H=20)				(|A|=178 747 880, H=19)			
	hccost	23/20	53011/54837	7	0.036	21/19	54861/49269	10	0.395	23/20	55650/49499	10	1.217	22/19	56431/49849	10	107.999
	hchops1	17/20	78191/52772	8	0.058	17/19	96797/56039	6	0.979	16/20	91037/50861	6	14.911	17/19	84997/55555	6	254.144
	hchops2	17/20	78191/52772	8	0.050	17/19	96797/56039	6	0.696	16/20	91037/50861	6	15.312	17/19	84997/55555	6	254.686
	hchops3	17/20	78191/52772	9	0.058	17/19	96797/61146	6	1.296	16/20	91037/50861	7	19.225	17/19	84997/49919	10	257.521
	hchops4	17/20	78191/52772	9	0.043	17/19	96797/61146	6	0.675	16/20	91037/50861	7	15.098	17/19	84997/49919	10	243.436
11		(|A|=44 489, H=20)				(|A|=642 770, H=20)				(|A|=8 738 433, H=20)				(|A|=178 747 112, H=19)			
	hccost	22/19	56676/52860	5	0.028	23/20	59241/50528	8	0.313	22/20	59107/51313	7	3.445	24/19	64194/51737	11	114.146
	hchops1	18/20	93420/53524	9	0.076	18/20	101408/57412	8	0.930	17/20	100771/50972	7	12.245	18/19	100628/54054	11	250.609
	hchops2	18/20	93420/53524	9	0.064	18/20	101408/57412	8	0.603	17/20	100771/50972	7	12.148	18/19	100628/54054	11	154.576
	hchops3	18/20	93420/53524	10	0.069	18/20	101408/57412	9	0.724	17/20	100771/50972	8	13.203	18/19	100628/57181	13	231.240
	hchops4	18/20	93420/53524	10	0.052	18/20	101408/57412	9	0.550	17/20	100771/50972	8	10.737	18/19	100628/57181	13	214.768

Table 4. Test Results: world=10, range=200, part 2

Seed		grid=12, |N|=809 h$_0$/h	c$_0$/c	o	Time	grid=14, |N|=3209 h$_0$/h	c$_0$/c	o	Time	grid=16, |N|=12509 h$_0$/h	c$_0$/c	o	Time	grid=18, |N|=55121 h$_0$/h	c$_0$/c	o	Time
0		(|A|=7 337, H=32)				(|A|=114 993, H=32)				(|A|=1 630 880, H=32)				(|A|=32 012 927, H=29)			
	hccost	34/32	191192/188229	2	0.018	38/32	204420/194234	9	0.168	36/32	207504/199062	9	1.125	32/29	155507/146465	8	10.449
	hchops1	31/32	247663/222908	4	0.029	30/32	246362/193078	8	0.149	31/32	258265/201959	12	1.799	26/29	186647/146627	13	27.547
	hchops2	31/32	247663/222908	4	0.020	30/32	246362/193078	8	0.117	31/31	258265/202684	9	1.967	26/29	186647/146627	13	19.724
	hchops3	31/32	247663/222419	7	0.020	30/32	246362/193078	7	0.102	31/31	258265/198723	6	1.134	26/29	186647/146627	14	18.364
	hchops4	31/32	247663/222419	7	0.016	30/32	246362/193078	7	0.097	31/31	258265/198723	6	1.223	26/29	186647/146627	14	15.451
1		(|A|=7 340, H=31)				(|A|=114 989, H=30)				(|A|=1 631 049, H=33)				(|A|=32 014 114, H=25)			
	hccost	35/30	205125/187501	4	0.013	37/29	206102/180613	8	0.114	36/32	187268/177527	8	0.531	28/25	134527/125444	8	7.577
	hchops1	29/31	234669/193219	8	0.031	30/30	239766/196026	4	0.115	29/33	232021/185507	6	1.293	24/25	162424/126197	9	18.727
	hchops2	29/31	234669/193219	8	0.026	30/30	239766/196026	4	0.080	29/33	232021/185507	6	1.091	24/25	162424/126197	9	7.572
	hchops3	29/31	234669/193219	9	0.025	30/30	239766/194194	6	0.071	29/33	232021/185507	7	0.702	24/25	162424/127137	13	6.535
	hchops4	29/31	234669/193219	9	0.024	30/30	239766/194194	6	0.065	29/33	232021/185507	7	0.614	24/25	162424/127137	13	6.251
2		(|A|=7 315, H=30)				(|A|=114 901, H=30)				(|A|=1 630 681, H=29)				(|A|=32 011 633, H=30)			
	hccost	31/30	177684/163872	2	0.013	35/30	187958/168423	4	0.042	33/29	181861/168470	6	0.792	35/30	177314/153729	9	8.525
	hchops1	28/28	218453/170297	3	0.018	28/30	224212/170202	4	0.166	27/29	212777/167484	9	1.490	27/30	217623/153724	10	28.388
	hchops2	28/28	218453/170297	3	0.016	28/30	224212/170202	4	0.117	27/29	212777/167484	9	1.222	27/30	217623/153724	10	19.348
	hchops3	28/28	218453/170297	4	0.012	28/30	224212/170202	5	0.100	27/29	212777/168803	6	0.634	27/30	217623/153724	11	16.228
	hchops4	28/28	218453/170297	4	0.008	28/30	224212/170202	5	0.073	27/29	212777/168803	6	0.556	27/30	217623/153724	11	14.692
3		(|A|=7 350, H=21)				(|A|=115 024, H=23)				(|A|=1 631 060, H=24)				(|A|=32 013 482, H=24)			
	hccost	21/21	118602/118602	0	0.005	25/22	134071/119833	6	0.028	27/24	133948/117491	4	0.249	27/23	127800/109913	12	6.395
	hchops1	20/21	123150/118602	2	0.011	24/22	165881/119833	7	0.115	24/24	172641/115095	11	1.522	24/23	164708/109913	12	24.815
	hchops2	20/21	123150/118602	2	0.008	24/22	165881/119833	9	0.107	24/24	172641/115095	11	0.979	24/24	164708/164708	0	1.281
	hchops3	20/21	123150/118602	2	0.006	24/22	165881/119833	7	0.067	24/24	172641/115861	13	0.508	24/23	164708/109913	19	10.776
	hchops4	20/21	123150/118602	2	0.007	24/22	165881/119833	7	0.046	24/24	172641/115861	13	0.517	24/23	164708/109913	19	10.976
4		(|A|=7 346, H=28)				(|A|=114 991, H=28)				(|A|=1 630 889, H=28)				(|A|=32 013 424, H=27)			
	hccost	33/28	175380/158841	5	0.020	29/28	164219/156313	4	0.052	30/28	171019/157224	7	0.630	31/26	159285/144188	13	13.071
	hchops1	27/28	194759/181762	3	0.024	26/28	203421/158812	3	0.121	27/26	195216/156333	9	1.428	25/27	176131/143613	10	32.880
	hchops2	27/28	194759/187883	2	0.014	26/28	203421/158812	3	0.101	27/26	195216/156333	9	0.926	25/27	176131/143613	10	19.018
	hchops3	27/28	194759/187883	2	0.010	26/28	203421/158812	3	0.077	27/26	195216/156327	10	0.771	25/27	176131/141771	12	14.501
	hchops4	27/28	194759/187883	2	0.010	26/28	203421/158812	3	0.062	27/26	195216/156327	10	0.730	25/27	176131/141771	12	13.988
5		(|A|=, H=)				(|A|=, H=)				(|A|=, H=)				(|A|=32 012 613, H=34)			
	hccost													37/34	193482/187108	8	21.567
	hchops1													30/34	245422/187631	7	53.727
	hchops2													30/34	245422/187901	6	40.839
	hchops3													30/34	245422/187901	7	32.139
	hchops4													30/34	245422/187901	7	28.164

Table 5. Test Results: world=11, range=100, part 1

Seed		grid=12, |N|=809				grid=14, |N|=3209				grid=16, |N|=12509				grid=18, |N|=55121			
		h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time
			(|A|=7 354, H=34)				(|A|=115 039, H=35)				(|A|=1 631 081, H=35)				(|A|=32 014 265, H=36)		
6	hccost	38/34	250910/247158	4	0.021	39/35	224053/221220	7	0.105	39/35	211598/205799	6	1.054	41/36	200244/194627	9	23.014
	hchops1	31/34	266849/245841	5	0.037	30/34	249951/228795	4	0.119	30/35	250173/215782	7	1.630	31/35	249921/208149	6	31.727
	hchops2	31/34	266849/245841	5	0.030	30/34	249951/228795	4	0.154	30/35	250173/215782	7	1.392	31/35	249921/208149	6	21.273
	hchops3	31/34	266849/245841	6	0.031	30/34	249951/228795	5	0.089	30/35	250173/215782	8	1.122	31/35	249921/208149	7	15.414
	hchops4	31/34	266849/245841	5	0.026	30/34	249951/228795	5	0.080	30/35	250173/215782	8	1.118	31/35	249921/208149	7	14.304
			(|A|=7 321, H=28)				(|A|=115 025, H=29)				(|A|=1 630 875, H=30)				(|A|=32 013 701, H=29)		
7	hccost	30/28	193844/188678	3	0.010	32/29	176339/163239	4	0.054	32/30	172272/160559	7	0.703	31/29	158202/149378	8	7.422
	hchops1	25/28	198775/188777	5	0.024	27/29	204716/162430	5	0.125	29/30	235252/165502	8	1.464	26/29	187939/155438	4	36.804
	hchops2	25/28	198775/188777	5	0.021	27/29	204716/162430	5	0.112	29/30	235252/165502	8	0.826	26/29	187939/155438	4	24.618
	hchops3	25/28	198775/188777	5	0.018	27/29	204716/162430	6	0.076	29/30	235252/165502	11	0.856	26/29	187939/155438	5	24.678
	hchops4	25/28	198775/188777	5	0.017	27/29	204716/162430	6	0.070	29/30	235252/165502	11	0.813	26/29	187939/155438	5	18.872
			(|A|=7 331, H=34)				(|A|=114 945, H=34)				(|A|=1 630 825, H=34)				(|A|=32 012 638, H=35)		
8	hccost	38/34	226573/226885	5	0.022	39/34	229743/214876	11	0.177	38/34	209283/205765	6	1.834	39/35	202817/181601	10	39.144
	hchops1	29/34	243579/232232	5	0.032	30/34	249829/221699	4	0.153	29/34	253494/219546	8	2.540	30/35	242226/195381	11	45.432
	hchops2	29/34	243579/232232	5	0.031	30/34	249829/221699	4	0.166	29/34	253494/219546	8	2.684	30/35	242226/195381	11	35.734
	hchops3	29/34	243579/232232	5	0.023	30/34	249829/221699	4	0.144	29/34	253494/219546	8	1.882	30/35	242226/195381	12	33.223
	hchops4	29/34	243579/232232	5	0.024	30/34	249829/221699	4	0.107	29/34	253494/219546	8	1.728	30/35	242226/195381	12	31.963
			(|A|=7 346, H=24)				(|A|=115 015, H=25)				(|A|=1 631 095, H=23)				(|A|=32 014 019, H=24)		
9	hccost	29/24	157730/149159	5	0.015	26/25	152833/137780	3	0.025	28/23	150940/136918	9	1.422	27/24	139455/116822	11	6.165
	hchops1	21/24	147127/143697	4	0.013	24/25	180408/142807	9	0.100	20/23	147533/137737	4	0.663	21/24	152477/116822	10	18.012
	hchops2	21/24	147127/143697	4	0.009	24/25	180408/142807	9	0.073	20/23	147533/137737	4	0.478	21/24	152477/116822	10	8.116
	hchops3	21/24	147127/143697	4	0.008	24/25	180408/142807	10	0.053	20/23	147533/137737	5	0.263	21/24	152477/116822	10	6.301
	hchops4	21/24	147127/143697	4	0.006	24/25	180408/142807	10	0.049	20/23	147533/137737	5	0.215	21/24	152477/116822	10	6.122
			(|A|=7 334, H=23)				(|A|=114 968, H=23)				(|A|=1 630 831, H=23)				(|A|=32 013 448, H=23)		
10	hccost	25/23	137771/136604	3	0.012	25/23	141875/120860	6	0.048	25/23	128687/122085	5	0.308	27/23	130711/109726	14	5.808
	hchops1	21/23	143765/136985	3	0.013	21/23	152156/131904	9	0.075	21/23	151991/122274	4	0.665	22/23	158722/110678	15	20.217
	hchops2	21/23	143765/136985	3	0.009	21/23	152156/131904	9	0.042	21/23	151991/122274	4	0.535	22/23	158722/110678	15	11.834
	hchops3	21/23	143765/136985	4	0.007	21/23	152156/131904	9	0.043	21/23	151991/122274	5	0.277	22/23	158722/111408	15	7.829
	hchops4	21/23	143765/136985	4	0.009	21/23	152156/131904	9	0.039	21/23	151991/122274	5	0.278	22/23	158722/111408	15	7.699
			(|A|=7 333, H=25)				(|A|=115 033, H=26)				(|A|=1 630 968, H=26)				(|A|=32 013 773, H=26)		
11	hccost	26/25	157277/156399	1	0.011	29/26	157498/146151	5	0.090	28/26	154000/146635	6	0.403	28/26	145048/138226	6	14.615
	hchops1	23/25	171707/163094	2	0.019	24/26	173977/143328	6	0.121	24/26	173013/148107	7	1.161	23/26	169144/134017	10	24.905
	hchops2	23/25	171707/163094	2	0.018	24/26	173977/145452	4	0.063	24/26	173013/148107	7	1.199	23/26	169144/134017	10	15.905
	hchops3	23/25	171707/163094	3	0.010	24/26	173977/145452	5	0.060	24/26	173013/146830	4	0.607	23/26	169144/134017	11	15.496
	hchops4	23/25	171707/163094	3	0.011	24/26	173977/145452	5	0.051	24/26	173013/146830	4	0.552	23/26	169144/134017	11	13.763

Table 6. Test Results: world=11, range=100, part 2

		grid=12, |N|=809				grid=14, |N|=3209				grid=16, |N|=12509				grid=18, |N|=55121			
Seed		h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time
0		(|A|=15 151, H=24)				(|A|=232 553, H=24)				(|A|=3 411 592, H=25)				(|A|=65 141 805, H=23)			
	hccost	29/24	83918/77454	3	0.026	28/24	80785/82015	7	0.243	29/25	88965/85536	8	2.605	28/23	70807/65730	7	34.886
	hchops1	23/24	121022/90196	5	0.032	23/24	139968/80833	7	0.315	22/25	149422/85022	6	5.838	22/23	119585/60202	11	66.688
	hchops2	23/24	121022/90196	5	0.020	23/24	139968/80833	7	0.240	22/25	149422/85022	6	4.168	22/23	119585/60202	11	39.351
	hchops3	23/24	121022/90196	8	0.028	23/24	139968/80833	7	0.243	22/25	149422/84720	8	3.664	22/23	119585/60202	11	44.322
	hchops4	23/24	121022/90196	8	0.024	23/24	139968/80833	7	0.207	22/25	149422/84720	8	3.113	22/23	119585/60202	11	36.294
1		(|A|=15 154, H=23)				(|A|=232 549, H=26)				(|A|=3 411 761, H=24)				(|A|=65 142 992, H=19)			
	hccost	26/23	78840/81666	4	0.026	28/26	89425/80065	5	0.079	28/24	84877/74310	9	1.596	24/19	60805/55648	6	27.051
	hchops1	21/23	106835/81397	9	0.051	23/23	130964/73609	10	0.254	23/24	131246/71603	11	5.203	18/19	91408/56335	8	62.675
	hchops2	21/23	106835/87134	4	0.027	23/23	130964/73609	10	0.219	23/24	131246/75700	8	1.774	18/19	91408/56335	8	34.619
	hchops3	21/23	106835/87134	4	0.028	23/23	130964/73609	11	0.181	23/24	131246/77166	11	1.879	18/19	91408/55785	11	31.817
	hchops4	21/23	106835/87134	4	0.026	23/23	130964/73609	11	0.162	23/24	131246/75700	8	1.287	18/19	91408/55785	11	25.613
2		(|A|=15 129, H=22)				(|A|=232 461, H=22)				(|A|=3 411 393, H=23)				(|A|=65 140 511, H=22)			
	hccost	23/22	78535/75434	1	0.009	24/22	77004/71379	5	0.152	27/23	82614/75300	5	2.278	27/22	77677/66923	10	81.726
	hchops1	21/22	119306/95883	2	0.021	20/22	119454/71379	6	0.333	19/23	113578/73608	8	4.187	19/22	113351/65268	7	78.171
	hchops2	21/22	119306/95883	2	0.024	20/22	119454/71379	6	0.288	19/23	113578/73608	8	2.670	19/22	113351/65268	7	49.523
	hchops3	21/22	119306/95883	3	0.016	20/22	119454/71379	7	0.267	19/23	113578/73608	9	2.885	19/22	113351/65295	6	45.143
	hchops4	21/22	119306/95883	3	0.012	20/22	119454/71379	7	0.207	19/23	113578/73608	9	2.113	19/22	113351/65268	7	57.101
3		(|A|=15 164, H=17)				(|A|=232 584, H=19)				(|A|=3 411 772, H=19)				(|A|=65 142 360, H=19)			
	hccost	18/17	54439/58574	1	0.010	22/18	68781/58753	8	0.060	23/19	63763/59173	4	1.037	23/19	64706/51345	10	17.370
	hchops1	15/17	70320/54758	3	0.018	18/18	89575/60473	6	0.216	17/19	87339/58032	7	2.917	18/19	88432/54588	7	44.292
	hchops2	15/17	70320/54758	3	0.012	18/18	89575/60473	6	0.164	17/19	87339/58032	7	1.635	18/19	88432/54588	7	21.254
	hchops3	15/17	70320/54758	3	0.013	18/19	89575/60182	6	0.092	17/19	87339/57267	7	1.637	18/18	88432/51145	8	31.453
	hchops4	15/17	70320/54758	3	0.012	18/19	89575/60182	7	0.086	17/19	87339/57267	7	1.300	18/18	88432/51145	8	27.676
4		(|A|=15 160, H=22)				(|A|=232 551, H=22)				(|A|=3 411 601, H=22)				(|A|=65 142 302, H=21)			
	hccost	23/22	71303/74586	2	0.017	26/22	74887/68053	3	0.077	25/22	80436/75351	7	1.777	26/21	71025/61581	12	40.600
	hchops1	20/22	103738/70730	5	0.031	22/21	113738/66959	13	0.309	21/22	124262/69690	7	3.924	19/21	101823/62957	6	64.035
	hchops2	20/22	103738/70730	5	0.020	22/21	113738/66959	13	0.401	21/22	124262/69690	7	2.462	19/21	101823/62957	6	34.365
	hchops3	20/22	103738/76601	4	0.016	22/22	113738/63550	13	0.206	21/22	124262/69690	7	2.035	19/21	101823/62033	10	52.404
	hchops4	20/22	103738/76601	4	0.015	22/22	113738/63550	13	0.209	21/22	124262/69690	7	1.897	19/21	101823/62033	10	53.253
5		(|A|=, H=)				(|A|=, H=)				(|A|=, H=)				(|A|=65 141 491, H=25)			
	hccost													30/25	83419/77357	7	52.895
	hchops1													22/25	139808/78215	8	98.579
	hchops2													22/25	139808/76363	15	96.130
	hchops3													22/25	139808/78277	9	73.401
	hchops4													22/25	139808/78277	9	65.035

Table 7. Test Results: world=11, range=200, part 1

		grid=12, |N|=809				grid=14, |N|=3209				grid=16, |N|=12509				grid=18, |N|=55121			
Seed		h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time
		(|A|=15 168, H=27)				(|A|=232 599, H=25)				(|A|=3 411 793, H=27)				(|A|=65 143 143, H=26)			
6	hccost	31/27	93257/82334	8	0.028	30/25	88575/84161	8	0.400	30/27	82248/80707	6	1.952	31/26	84501/78306	15	87.876
	hchops1	23/27	136306/89563	8	0.059	22/25	151018/84161	7	0.478	25/27	164891/86985	9	6.899	22/26	137163/76314	7	102.656
	hchops2	23/27	136306/89563	8	0.037	22/25	151018/84161	7	0.644	25/27	164891/86985	9	4.578	22/26	137163/76314	7	63.339
	hchops3	23/27	136306/89563	9	0.029	22/25	151018/84161	7	0.427	25/27	164891/86979	7	3.516	22/26	137163/76314	8	70.328
	hchops4	23/27	136306/89563	9	0.028	22/25	151018/84161	7	0.348	25/27	164891/86985	9	3.570	22/26	137163/76314	8	69.966
		(|A|=15 135, H=22)				(|A|=232 585, H=23)				(|A|=3 411 587, H=23)				(|A|=65 142 579, H=23)			
7	hccost	26/22	80376/79611	6	0.030	25/23	83066/73961	5	0.097	25/23	83381/82023	5	1.577	26/23	70700/68282	5	34.534
	hchops1	19/22	114402/84434	8	0.029	19/23	110006/69985	6	0.437	20/23	118369/72253	15	6.399	22/23	129490/65878	8	79.986
	hchops2	19/22	114402/84434	8	0.025	19/23	110006/69985	6	0.645	20/23	118369/72253	15	4.976	22/23	129490/65878	8	39.782
	hchops3	19/22	114402/84434	8	0.025	19/23	110006/69985	7	0.451	20/23	118369/72253	16	4.284	22/23	129490/65270	10	60.108
	hchops4	19/22	114402/84434	8	0.022	19/23	110006/69985	7	0.324	20/23	118369/72253	16	3.673	22/23	129490/65270	10	55.950
		(|A|=15 145, H=24)				(|A|=232 505, H=25)				(|A|=3 411 537, H=26)				(|A|=65 141 516, H=25)			
8	hccost	28/24	93648/85559	10	0.044	30/24	104628/90316	13	0.311	29/26	100292/90352	9	2.124	30/25	86333/75685	11	61.895
	hchops1	22/24	137486/85559	8	0.041	23/25	162539/88619	13	0.433	23/26	150899/83383	8	4.030	22/25	130923/75685	11	94.815
	hchops2	22/24	137486/85559	8	0.031	23/25	162539/84829	8	0.445	23/26	150899/84869	7	2.691	22/25	130923/75685	11	64.965
	hchops3	22/24	137486/92841	7	0.032	23/25	162539/88941	8	0.251	23/26	150899/84869	7	2.158	22/25	130923/75685	12	61.842
	hchops4	22/24	137486/92841	7	0.029	23/25	162539/88941	7	0.220	23/26	150899/84869	7	1.979	22/25	130923/75685	12	61.824
		(|A|=15 160, H=20)				(|A|=232 575, H=20)				(|A|=3 411 807, H=20)				(|A|=65 142 897, H=20)			
9	hccost	21/20	64890/63640	2	0.007	22/19	78233/72895	8	0.129	24/20	71571/63355	8	1.328	24/20	64459/54810	10	28.711
	hchops1	17/19	75659/61335	5	0.019	17/20	103883/68050	8	0.223	17/20	114686/66988	5	3.164	18/20	91852/59169	9	60.946
	hchops2	17/19	75659/61335	5	0.012	17/20	103883/68050	8	0.267	17/20	114686/66988	5	2.032	18/20	91852/59169	9	23.016
	hchops3	17/19	75659/61335	6	0.012	17/20	103883/68050	8	0.144	17/20	114686/66988	5	2.149	18/20	91852/59230	9	29.124
	hchops4	17/19	75659/61335	6	0.012	17/20	103883/68050	8	0.145	17/20	114686/66988	5	1.500	18/20	91852/59230	9	27.392
		(|A|=15 148, H=19)				(|A|=232 528, H=19)				(|A|=3 411 543, H=19)				(|A|=65 142 326, H=19)			
10	hccost	20/19	62760/61680	3	0.010	20/19	73896/58827	8	0.052	21/19	71147/62876	6	0.498	23/18	68035/56644	9	24.342
	hchops1	18/19	88539/61680	7	0.031	17/17	100018/62211	9	0.193	19/18	111320/68059	9	2.567	20/19	114632/55507	11	68.942
	hchops2	18/19	88539/61680	7	0.022	17/17	100018/62211	9	0.138	19/18	111320/68059	9	1.306	20/19	114632/55507	11	38.549
	hchops3	18/19	88539/63976	8	0.016	17/18	100018/66123	6	0.119	19/17	111320/65173	19	2.223	20/18	114632/55598	8	37.221
	hchops4	18/19	88539/63976	8	0.015	17/18	100018/66123	6	0.084	19/17	111320/65173	19	2.101	20/18	114632/55598	8	32.723
		(|A|=15 147, H=21)				(|A|=232 593, H=20)				(|A|=3 411 680, H=21)				(|A|=65 142 651, H=21)			
11	hccost	23/21	76600/68062	5	0.019	25/20	71376/64578	7	0.202	25/21	68659/65053	8	1.263	24/21	67777/55159	7	22.079
	hchops1	20/21	110454/72743	7	0.029	18/20	96479/63313	8	0.239	18/20	97927/66412	6	4.573	20/21	105060/60149	13	65.134
	hchops2	20/21	110454/72743	7	0.024	18/20	96479/63313	8	0.197	18/20	97927/66412	6	3.565	20/21	105060/60149	13	32.883
	hchops3	20/21	110454/72743	7	0.024	18/20	96479/63313	9	0.193	18/20	97927/66412	7	3.322	20/20	105060/54138	17	40.783
	hchops4	20/21	110454/72743	7	0.022	18/20	96479/63313	9	0.145	18/20	97927/66412	7	2.819	20/20	105060/54138	17	38.618

Table 8. Test Results: world=11, range=200, part 2

Seed		grid=12, |N|=809				grid=14, |N|=3209				grid=16, |N|=12509				grid=18, |N|=55121			
		h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time
0			(|A|=8 303, H=24)				(|A|=134 586, H=21)				(|A|=1 915 240, H=22)				(|A|=38 902 410, H=21)		
	hccost	24/24	148646/148646	0	0.009	24/21	134930/119363	5	0.033	24/21	128924/115893	4	0.539	24/21	129369/120391	5	32.556
	hchops1	22/22	158073/142467	2	0.020	21/21	152438/118617	6	0.143	20/21	138572/116637	7	1.269	19/21	141725/118155	4	42.512
	hchops2	22/22	158073/142467	2	0.010	21/21	152438/119363	6	0.067	20/21	138572/116637	7	0.770	19/21	141725/118155	4	27.512
	hchops3	22/22	158073/142467	2	0.011	21/21	152438/119363	8	0.102	20/21	138572/116637	7	0.814	19/21	141725/118155	4	21.320
	hchops4	22/22	158073/142467	2	0.010	21/21	152438/119363	8	0.081	20/21	138572/116637	7	0.749	19/21	141725/118155	4	20.604
1			(|A|=8 274, H=30)				(|A|=134 579, H=26)				(|A|=1 905 809, H=27)				(|A|=38 460 428, H=26)		
	hccost	30/30	181007/181007	0	0.010	28/26	159238/148085	7	0.072	31/27	164619/147918	7	0.803	29/26	157150/146177	9	46.622
	hchops1	30/30	219229/189507	4	0.022	26/26	184411/147064	6	0.167	27/27	202594/148105	13	1.933	25/25	188047/145511	10	70.211
	hchops2	30/30	219229/189507	4	0.012	26/26	184411/147064	6	0.064	27/27	202594/148105	13	1.203	25/26	188047/146464	8	27.654
	hchops3	30/30	219229/189507	6	0.013	26/26	184411/147064	6	0.058	27/27	202594/147762	11	1.189	25/26	188047/147440	8	30.069
	hchops4	30/30	219229/189507	6	0.014	26/26	184411/147064	6	0.058	27/27	202594/147762	11	1.052	25/26	188047/147440	8	32.559
2			(|A|=8 613, H=35)				(|A|=135 584, H=31)				(|A|=1 937 327, H=32)				(|A|=39 119 813, H=33)		
	hccost	39/35	227330/219267	8	0.044	34/31	205496/191641	8	0.137	37/32	201565/191203	6	1.395	35/33	205081/191143	6	44.567
	hchops1	33/35	263309/219267	7	0.028	30/31	235799/192701	6	0.214	29/32	243551/195295	4	1.773	30/33	249401/184338	8	89.354
	hchops2	33/35	263309/230218	5	0.022	30/31	235799/201631	4	0.073	29/32	243551/195295	4	1.193	30/32	249401/183561	6	42.484
	hchops3	33/35	263309/230218	6	0.019	30/31	235799/206578	5	0.080	29/32	243551/195295	4	1.126	30/33	249401/184338	8	58.510
	hchops4	33/35	263309/230218	6	0.017	30/31	235799/206578	5	0.065	29/32	243551/195295	4	1.038	30/33	249401/184338	8	52.611
3			(|A|=8 921, H=29)				(|A|=136 864, H=27)				(|A|=1 938 561, H=26)				(|A|=39 344 413, H=24)		
	hccost	31/29	160818/152624	4	0.016	29/27	160292/147825	6	0.099	30/25	160259/147542	15	1.719	29/24	143504/128612	16	39.059
	hchops1	29/29	227472/152626	6	0.028	25/27	183562/145711	6	0.247	24/26	176898/150866	8	1.525	21/24	156897/128660	6	46.437
	hchops2	29/29	227472/152626	6	0.020	25/27	183562/145711	6	0.109	24/26	176898/150866	8	0.951	21/24	156897/128660	6	23.696
	hchops3	29/29	227472/152624	7	0.028	25/27	183562/145711	6	0.113	24/26	176898/150866	9	1.113	21/24	156897/128660	7	28.113
	hchops4	29/29	227472/152624	7	0.024	25/27	183562/145711	6	0.082	24/26	176898/150866	9	0.989	21/24	156897/128660	7	21.370
4			(|A|=8 488, H=33)				(|A|=137 876, H=33)				(|A|=1 913 607, H=33)				(|A|=38 614 605, H=32)		
	hccost	37/33	224619/215307	3	0.021	36/33	224259/213724	4	0.108	37/33	214358/207284	7	1.455	37/32	207622/191280	8	32.498
	hchops1	30/33	248937/234770	4	0.023	31/33	271734/203909	10	0.270	30/31	271951/194808	7	1.899	29/32	249262/211292	5	55.186
	hchops2	30/33	248937/234770	4	0.014	31/33	271734/203909	10	0.142	30/31	271951/194808	7	1.116	29/32	249262/211292	5	29.509
	hchops3	30/33	248937/234770	4	0.017	31/33	271734/203909	12	0.199	30/31	271951/194808	8	1.373	29/32	249262/211292	7	40.747
	hchops4	30/33	248937/234770	4	0.014	31/33	271734/203909	11	0.147	30/31	271951/194808	8	1.115	29/32	249262/211292	7	31.575
5			(|A|=8 470, H=25)				(|A|=138 066, H=24)				(|A|=1 961 613, H=24)				(|A|=39 547 118, H=24)		
	hccost	26/25	159976/154377	1	0.010	26/24	145603/133728	7	0.071	25/24	141816/136206	4	0.606	26/23	145864/133612	5	17.283
	hchops1	24/23	164902/152061	3	0.015	22/24	154129/133623	6	0.131	22/24	164975/139554	5	1.214	23/24	170229/133446	8	54.082
	hchops2	24/23	164902/152061	3	0.010	22/24	154129/133623	6	0.053	22/24	164975/139554	5	0.705	23/24	170229/133446	8	19.255
	hchops3	24/23	164902/152061	3	0.009	22/24	154129/133623	6	0.058	22/24	164975/139554	5	0.766	23/24	170229/133697	8	21.577
	hchops4	24/23	164902/152061	3	0.009	22/24	154129/133623	6	0.050	22/24	164975/139554	5	0.688	23/24	170229/133697	8	16.908

Table 9. Test Results: world=12, range=100, part 1

		grid=12, |N|=809				grid=14, |N|=3209				grid=16, |N|=12509				grid=18, |N|=55121			
Seed		h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time
6			(|A|=8 784, H=30)				(|A|=138 057, H=26)				(|A|=1 929 458, H=26)				(|A|=39 605 018, H=27)		
	hccost	31/30	181403/178271	2	0.018	28/26	166399/146944	13	0.108	29/26	154933/141068	8	0.785	29/27	158463/143590	5	13.235
	hchops1	27/30	209655/181640	4	0.020	25/26	184003/147666	10	0.188	25/26	187469/141951	14	2.028	25/26	191909/138517	15	57.908
	hchops2	27/30	209655/181640	4	0.013	25/26	184003/147666	9	0.068	25/26	187469/141951	14	1.273	25/26	191909/138517	15	28.704
	hchops3	27/30	209655/181640	5	0.012	25/26	184003/147666	10	0.082	25/26	187469/143341	10	1.031	25/26	191909/138517	17	34.474
	hchops4	27/30	209655/181640	5	0.011	25/26	184003/147666	10	0.066	25/26	187469/143341	10	0.934	25/26	191909/138517	17	26.672
7			(|A|=8 547, H=24)				(|A|=140 152, H=24)				(|A|=1 979 980, H=23)				(|A|=39 951 099, H=24)		
	hccost	25/23	145994/136232	4	0.012	27/24	143897/127430	10	0.055	27/22	143071/123838	6	0.621	26/22	133824/119991	14	24.103
	hchops1	22/24	164586/143309	5	0.016	23/24	155912/129317	10	0.145	22/23	164259/124794	7	1.240	22/23	157603/121450	13	43.667
	hchops2	22/24	164586/143309	5	0.012	23/24	155912/129317	10	0.064	22/23	164259/124794	7	0.660	22/23	157603/121450	13	18.710
	hchops3	22/24	164586/143309	5	0.010	23/24	155912/129317	11	0.072	22/23	164259/124794	8	0.690	22/23	157603/121450	14	20.674
	hchops4	22/24	164586/143309	5	0.009	23/24	155912/129317	11	0.059	22/23	164259/124794	8	0.619	22/23	157603/121450	14	18.556
8			(|A|=7 997, H=24)				(|A|=140 801, H=23)				(|A|=1 910 359, H=22)				(|A|=38 858 500, H=24)		
	hccost	26/24	147868/145756	5	0.021	26/23	134636/118037	8	0.052	24/22	128762/116386	8	0.727	26/23	132441/119051	5	9.233
	hchops1	23/24	168486/153166	3	0.017	22/22	147397/116438	7	0.171	20/22	136525/118045	7	1.045	22/24	160622/128839	6	44.534
	hchops2	23/24	168486/153166	3	0.011	22/22	147397/116438	8	0.063	20/22	136525/118045	7	0.556	22/24	160622/128839	6	23.505
	hchops3	23/24	168486/153166	4	0.010	22/22	147397/116438	8	0.066	20/22	136525/118045	7	0.550	22/24	160622/129226	7	33.522
	hchops4	23/24	168486/153166	4	0.010	22/22	147397/116438	8	0.056	20/22	136525/118045	7	0.524	22/24	160622/129226	7	22.820
9			(|A|=8 425, H=20)				(|A|=138 245, H=20)				(|A|=1 962 292, H=20)				(|A|=39 517 535, H=19)		
	hccost	22/20	110566/106409	2	0.011	24/20	124874/102304	6	0.043	24/20	123246/105736	8	0.935	22/19	114947/102792	4	10.302
	hchops1	19/19	119093/110974	2	0.012	19/18	119637/114360	2	0.098	18/20	117868/104320	7	1.090	18/18	126335/114253	3	47.798
	hchops2	19/19	119093/110974	2	0.007	19/18	119637/114360	2	0.031	18/20	117868/104320	7	0.632	18/18	126335/114253	3	13.649
	hchops3	19/19	119093/110974	2	0.006	19/18	119637/114360	2	0.039	18/20	117868/104320	7	0.639	18/18	126335/114253	3	19.990
	hchops4	19/19	119093/110974	2	0.008	19/18	119637/114360	2	0.031	18/20	117868/104320	7	0.666	18/18	126335/114253	3	13.026
10			(|A|=8 146, H=29)				(|A|=134 745, H=26)				(|A|=1 896 430, H=27)				(|A|=38 424 026, H=26)		
	hccost	29/29	186204/186204	0	0.009	28/26	158811/153538	3	0.105	31/27	175298/153674	15	1.391	29/26	161092/154705	8	48.042
	hchops1	26/29	208374/186205	5	0.026	24/26	184317/164419	3	0.236	25/27	195477/164919	4	1.596	24/26	189710/150583	9	62.415
	hchops2	26/29	208374/186205	5	0.019	24/26	184317/164419	3	0.126	25/27	195477/164919	4	0.942	24/26	189710/150583	9	43.064
	hchops3	26/29	208374/186205	5	0.020	24/26	184317/164419	4	0.165	25/27	195477/164919	5	0.966	24/26	189710/150583	10	39.963
	hchops4	26/29	208374/186205	5	0.019	24/26	184317/164419	4	0.120	25/27	195477/164919	5	0.875	24/26	189710/150583	10	38.386
11			(|A|=8 697, H=31)				(|A|=138 626, H=31)				(|A|=1 916 637, H=31)				(|A|=39 139 870, H=31)		
	hccost	34/31	226823/205575	6	0.030	34/31	195088/188955	4	0.109	36/31	204761/187892	11	1.605	33/31	182340/173198	10	38.491
	hchops1	30/29	249274/217442	3	0.024	29/29	237906/228744	3	0.139	28/31	232285/197002	6	1.782	29/31	243146/188223	5	74.930
	hchops2	30/31	249274/244170	1	0.012	29/29	237906/228744	3	0.068	28/31	232285/197002	6	1.141	29/31	243146/188223	5	33.754
	hchops3	30/31	249274/244170	1	0.010	29/29	237906/228744	3	0.074	28/31	232285/197002	7	1.237	29/31	243146/183135	6	52.686
	hchops4	30/31	249274/244170	1	0.011	29/29	237906/228744	3	0.060	28/31	232285/197002	7	1.155	29/31	243146/188223	5	31.575

Table 10. Test Results: world=12, range=100, part 2

Seed		grid=12, |N|=809				grid=14, |N|=3209				grid=16, |N|=12509				grid=18, |N|=55121			
		h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time
		(|A|=23 762, H=20)				(|A|=380 690, H=17)				(|A|=5 585 245, H=17)				(|A|=111 071 230, H=17)			
0	hccost	23/20	65172/61545	4	0.025	20/17	64728/54058	5	0.232	20/17	64477/55365	8	5.486	19/17	61304/51645	5	144.847
	hchops1	17/20	86764/57879	5	0.031	15/17	71293/54058	4	0.316	16/17	84841/53202	6	8.805	17/17	101392/51645	8	323.480
	hchops2	17/20	86764/57879	5	0.020	15/17	71293/54058	4	0.194	16/17	84841/53202	4	5.070	17/17	101392/52285	7	241.263
	hchops3	17/20	86764/57879	5	0.018	15/17	71293/54058	4	0.258	16/17	84841/53202	4	5.020	17/17	101392/51645	9	391.602
	hchops4	17/20	86764/57879	5	0.020	15/17	71293/54058	4	0.188	16/17	84841/53202	4	4.868	17/17	101392/51645	9	271.341
		(|A|=23 562, H=21)				(|A|=379 734, H=20)				(|A|=5 535 653, H=21)				(|A|=109 501 823, H=20)			
1	hccost	24/21	73605/69184	8	0.043	24/20	65702/65360	6	0.163	25/21	69814/60477	6	3.388	23/20	62686/60341	7	179.209
	hchops1	19/21	96013/70269	5	0.048	17/20	101693/65307	6	0.397	20/21	121255/60326	13	9.134	19/20	102974/60311	8	478.571
	hchops2	19/21	96013/67884	5	0.021	17/20	101693/65307	6	0.281	20/21	121255/60326	13	5.566	19/20	102974/60311	8	235.710
	hchops3	19/21	96013/67884	4	0.018	17/20	101693/65307	6	0.427	20/21	121255/62305	8	7.069	19/20	102974/60311	10	286.907
	hchops4	19/21	96013/67884	4	0.018	17/20	101693/65307	6	0.360	20/21	121255/62305	8	6.671	19/20	102974/60311	10	234.301
		(|A|=24 910, H=24)				(|A|=388 126, H=24)				(|A|=5 694 653, H=24)				(|A|=112 574 251, H=24)			
2	hccost	29/24	82823/79325	8	0.044	28/24	80142/72715	10	0.300	28/24	77036/69748	12	7.036	29/24	76621/70358	12	207.461
	hchops1	22/24	120921/79575	7	0.069	21/24	110281/72589	10	0.397	20/24	123077/71213	7	13.358	21/24	141811/68290	20	463.137
	hchops2	22/24	120921/79575	7	0.045	21/24	110281/72589	10	0.263	20/24	123077/71213	7	9.606	21/24	141811/68290	20	321.014
	hchops3	22/24	120921/79575	8	0.046	21/24	110281/72589	11	0.357	20/24	123077/71213	9	14.369	21/24	141811/69418	12	331.369
	hchops4	22/24	120921/79575	8	0.039	21/24	110281/72589	11	0.319	20/24	123077/71213	9	13.868	21/24	141811/69418	12	257.478
		(|A|=25 886, H=20)				(|A|=391 744, H=21)				(|A|=5 708 244, H=20)				(|A|=113 199 398, H=19)			
3	hccost	24/20	67743/63654	7	0.027	24/21	66875/61136	12	0.351	24/20	64275/59717	7	3.823	22/19	58573/54506	10	150.366
	hchops1	18/20	87614/63654	9	0.044	17/21	102074/61136	8	0.462	18/20	101362/60969	8	13.801	18/19	113102/54506	10	413.643
	hchops2	18/20	87614/63654	9	0.030	17/21	102074/61136	8	0.338	18/20	101362/60969	11	9.087	18/19	113102/54506	10	287.814
	hchops3	18/20	87614/63654	9	0.036	17/21	102074/61136	9	0.512	18/20	101362/60969	11	11.661	18/19	113102/66871	7	301.884
	hchops4	18/20	87614/63654	9	0.033	17/21	102074/61136	9	0.374	18/20	101362/60969	11	10.795	18/19	113102/66871	7	218.351
		(|A|=24 561, H=25)				(|A|=394 212, H=25)				(|A|=5 573 768, H=24)				(|A|=110 530 653, H=23)			
4	hccost	30/25	86017/77729	11	0.033	30/25	81617/74622	11	0.611	30/24	84294/77176	9	9.558	31/23	82075/72508	10	322.609
	hchops1	22/25	125521/76839	11	0.066	22/25	139583/73599	9	0.651	21/24	141987/72771	8	15.273	21/23	145260/72508	9	443.162
	hchops2	22/25	125521/86272	5	0.032	22/25	139583/72993	10	0.476	21/24	141987/72771	8	10.802	21/23	145260/72508	8	290.319
	hchops3	22/25	125521/86272	6	0.033	22/25	139583/82070	8	0.591	21/24	141987/72771	7	9.835	21/23	145260/75547	11	398.826
	hchops4	22/25	125521/86272	6	0.032	22/25	139583/82070	8	0.490	21/24	141987/72771	7	8.391	21/23	145260/75547	11	303.969
		(|A|=24 275, H=19)				(|A|=392 287, H=20)				(|A|=5 709 149, H=19)				(|A|=112 825 633, H=19)			
5	hccost	22/19	66109/62979	8	0.052	22/20	67367/60926	3	0.178	25/19	72379/56811	7	2.916	24/18	68357/58113	9	287.235
	hchops1	18/19	91790/60186	9	0.032	18/20	96545/62011	5	0.478	18/19	97968/62639	7	12.121	17/19	94664/56347	7	358.133
	hchops2	18/19	91790/60186	9	0.027	18/20	96545/62011	5	0.277	18/19	97968/62639	7	8.096	17/19	94664/56347	7	245.155
	hchops3	18/18	91790/63639	9	0.025	18/20	96545/62011	6	0.379	18/19	97968/57611	11	11.085	17/19	94664/56347	7	217.239
	hchops4	18/18	91790/63639	9	0.026	18/20	96545/62011	6	0.326	18/19	97968/57611	11	11.569	17/19	94664/56347	7	174.541

Table 11. Test Results: world=12, range=200, part 1

Seed		grid=12, |N|=809				grid=14, |N|=3209				grid=16, |N|=12509				grid=18, |N|=55121			
		h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time	h_0/h	c_0/c	o	Time
6		(|A|=25 431, H=22)				(|A|=393 594, H=21)				(|A|=5 642 607, H=22)				(|A|=113 470 443, H=21)			
	hccost	25/22	75025/67570	4	0.016	26/21	71875/64080	11	0.200	27/22	69936/60269	7	5.027	24/21	65711/58717	9	106.385
	hchops1	21/22	109298/68050	7	0.037	18/21	100378/62870	7	0.395	20/22	115247/59344	9	13.666	20/21	117815/60631	12	288.991
	hchops2	21/22	109298/68050	7	0.024	18/21	100378/62870	7	0.228	20/22	115247/59344	9	13.831	20/21	117815/60631	12	182.724
	hchops3	21/22	109298/68050	8	0.024	18/21	100378/62870	8	0.360	20/22	115247/58605	11	9.042	20/21	117815/64314	12	272.272
	hchops4	21/22	109298/68050	8	0.022	18/21	100378/62870	8	0.266	20/22	115247/58605	11	8.315	20/21	117815/64314	12	171.529
7		(|A|=24 698, H=18)				(|A|=398 940, H=18)				(|A|=5 882 066, H=18)				(|A|=116 197 908, H=18)			
	hccost	20/17	71701/64704	5	0.027	20/18	63689/61370	6	0.254	22/18	68485/57891	5	4.685	22/18	63868/55585	6	133.623
	hchops1	16/17	86680/64704	6	0.022	17/18	99968/61370	7	0.580	17/18	87185/57891	10	10.106	16/18	91507/58297	6	169.091
	hchops2	16/17	86680/64704	6	0.018	17/18	99968/61370	7	0.283	17/18	87185/57891	10	8.960	16/18	91507/58297	6	74.669
	hchops3	16/17	86680/64704	7	0.019	17/18	99968/61370	7	0.437	17/18	87185/61329	9	5.648	16/18	91507/56603	18	262.505
	hchops4	16/17	86680/64704	7	0.017	17/18	99968/61370	7	0.345	17/18	87185/61329	9	4.850	16/18	91507/56603	18	171.021
8		(|A|=22 624, H=18)				(|A|=394 290, H=18)				(|A|=5 442 005, H=18)				(|A|=108 834 847, H=18)			
	hccost	20/18	60667/58210	4	0.017	21/18	61145/51326	5	0.162	21/18	60177/50482	7	4.833	21/18	60764/50476	9	196.485
	hchops1	17/18	91566/58366	5	0.031	18/18	93736/53766	9	0.348	16/18	91803/51568	4	8.843	17/18	99992/53166	12	515.269
	hchops2	17/18	91566/58366	5	0.023	18/18	93736/53766	9	0.219	16/18	91803/51568	4	6.796	17/18	99992/53166	12	261.169
	hchops3	17/18	91566/58055	6	0.031	18/18	93736/55668	4	0.214	16/18	91803/51568	5	6.812	17/18	99992/53309	12	288.973
	hchops4	17/18	91566/58055	6	0.023	18/18	93736/55668	4	0.159	16/18	91803/51568	5	5.717	17/18	99992/53166	12	214.741
9		(|A|=24 275, H=16)				(|A|=391 411, H=17)				(|A|=5 734 534, H=16)				(|A|=112 863 331, H=16)			
	hccost	19/16	51843/46487	4	0.013	19/17	54005/50400	3	0.283	19/16	53463/49615	4	3.750	19/16	53077/47182	9	287.612
	hchops1	16/16	69142/46487	4	0.034	15/17	82368/51888	4	0.358	16/16	81706/51039	5	12.880	15/16	81246/48672	5	286.049
	hchops2	16/16	69142/46487	4	0.019	15/17	82368/51888	4	0.272	16/16	81706/51039	4	7.639	15/16	81246/48672	5	149.860
	hchops3	16/16	69142/48026	5	0.023	15/17	82368/51888	5	0.391	16/16	81706/51039	4	11.429	15/16	81246/48672	5	197.014
	hchops4	16/16	69142/48026	5	0.021	15/17	82368/51888	5	0.277	16/16	81706/51039	4	10.248	15/16	81246/48672	5	137.406
10		(|A|=23 218, H=21)				(|A|=381 152, H=20)				(|A|=5 547 754, H=21)				(|A|=110 236 445, H=21)			
	hccost	24/21	71140/70396	6	0.030	23/19	70487/67880	9	0.390	24/21	79163/68909	10	5.453	23/21	73127/62898	14	270.752
	hchops1	18/21	104404/68166	9	0.057	18/20	110720/65262	8	0.495	19/21	115895/62711	9	12.200	18/21	118554/62987	7	370.300
	hchops2	18/21	104404/68166	9	0.041	18/20	110720/65262	8	0.351	19/21	115895/62711	8	9.534	18/21	118554/64428	5	230.997
	hchops3	18/21	104404/68885	6	0.038	18/20	110720/65262	9	0.520	19/21	115895/67268	6	10.534	18/21	118554/64428	6	195.051
	hchops4	18/21	104404/68885	6	0.032	18/20	110720/65262	9	0.392	19/21	115895/67268	6	8.651	18/21	118554/64428	6	181.835
11		(|A|=25 147, H=23)				(|A|=393 524, H=23)				(|A|=5 580 932, H=23)				(|A|=111 704 300, H=23)			
	hccost	28/23	81191/78129	6	0.044	28/23	77893/70788	10	0.293	28/23	81103/66997	13	6.097	27/23	77293/63642	10	168.927
	hchops1	21/23	112331/77383	9	0.053	20/23	128673/69143	6	0.541	22/23	148378/69962	12	13.590	20/23	131964/66192	11	372.610
	hchops2	21/23	112331/77383	9	0.039	20/23	128673/69397	6	0.369	22/23	148378/69962	12	7.968	20/23	131964/66192	11	248.675
	hchops3	21/23	112331/79066	9	0.030	20/23	128673/69397	7	0.456	22/23	148378/69302	11	9.232	20/23	131964/69475	10	185.137
	hchops4	21/23	112331/79066	9	0.029	20/23	128673/69397	7	0.423	22/23	148378/69302	11	9.602	20/23	131964/69475	10	160.031

Table 12. Test Results: world=12, range=200, part 2