# A replication of: 'A variable neighborhood search method for the orienteering problem with hotel selection' by Divsalar et al. (2013)

Erasmus University Rotterdam
Erasmus School of Economics
Bachelor thesis: Econometrics and Operational Research

Name Student: Ruud Moers
Student ID number: 370376

Supervisor: Dr. T.A.B. Dollevoet
Second Assessor: DR. F. Frasincar

July 4, 2016

**Abstract**

In this paper I tried to reproduce and improve the methods and results by Divsalar et al. (2013) for solving numerous OPHS instances. I added small adjustments to the metaheuristic algorithm proposed by Divsalar et al. (2013) and implemented a new improvement method "Trips-Shake", On average I was unfortunately unable to beat the results by Divsalar et al. (2013), and also the computation times I generated greatly exceeded those by Divsalar et al. (2013). However, on several OPHS instances I was able to get better results, where in particular for the smaller OPHS instances.

## 1 Introduction

The orienteering problem with hotel selection (OPHS) can be viewed as an extension on the classic orienteering problem (OP). The OP can be described as a tourist planning the most satisfying tour, knowing the hotel he leaves at the start and the hotel he arrives at the end. The satisfaction of the traveler depends on the tourist attractions, or hotspots, he visits during his tour. Obviously, the total duration of the tour is limited. The traveler's preferences differ per hotspot, therefore optimizing the total number of hotspots visited is not per definition coherent to finding the tour with the optimal utility. Various methods to solve such an OP are offered by Vansteenwegen et al. (2011).

An OPHS can be described similarly to the OP. However, in an OPHS the tour of the tourist is divided into day trips. At the end of each trip the tourist has to arrive in a hotel, from which he will depart again the next day. Once more only the hotels at which he will start and end his tour are specified. For each day trip the tourist has a maximum amount of time he wants to travel. The tourist is possible to spend the night in any hotel located in the traveling area, which can be reached without violating his time constraints. Since visiting the same hotspot for the second time in the tour does not increase the total utility of the tourist, the optimal utility of each day trip depends on the other day trips. Therefore it is unsatisfying to solely evaluate all day trips as individual OPs for every possible hotel pair. This aspect makes an OPHS in general drastically more complex than an classic OP.

Divsalar et al. (2013) found that solving an OPHS using established optimizing techniques require long computation times, which increase rapidly when the number of day trips, hotspots and hotels increase. Therefore they introduced a metaheuristic algorithm to solve the OPHS faster. However, this algorithm is unable to guarantee optimality. The algorithm was implemented in preparated OPHSs. Computation times dropped greatly in comparison to the established optimizing techniques and the number of times the actual optimal solution was found was impressive. In this paper I replicated the metaheuristic algorithm introduced by Divsalar et al. (2013) with some adjustments and extensions and tried to reproduce similar results. Since I will refer multiple times to the paper of Divsalar et al.(2013), I will refer to the paper as DVC in the rest of the paper.

## 2 OPHS Formulation

To clearly explain an individual OPHS, a numerous amount of information needs to be defined. This section clarifies all indices and parameters that are used to discuss and finally solve an OPHS.

As described earlier, the tourist has the option to decide which hotels and hotspots he visits during his tour. Obviously, these hotels and hotspots are specified beforehand. Let $H$ be the number of hotels and $N$ be the number of hotspots present in the traveling region. Both a hotel and a hotspot represent a location and can both be observed as a vertex on a map. Let vertex $i$ represent a hotel for $i = 1, ..., H$, and a hotspot for $i = H + 1, ..., H + N$, where $i = 1$ and $i = 2$ represent the first hotel and the last hotel in the tour respectively. The tourist is allowed to visit these two hotel multiple times during his tour like the other hotels. Each hotspot has its own score $S_i$ to reflect the preferences of the tourist. The time it takes to travel from vertex $i$ to vertex $j$ is defined by $t_{ij}$. Let D be the number of day trips the tourist plans to make and $d$ represent the day at which a trip takes place, where $d = 1, ..., D$. The total length of the tour is bounded by $T$ and each individual day trip by $T_d$. The final utility of the tour is measured by the sum of the scores of all hotspots that are included in the tour. Each vertex is included at most once in the tour. The purpose of the OPHS is to maximize this final utility.

## 3 Proposed Algorithms

In this section, multiple metaheuristic algorithms I used to solve the OPHS are described. First, the skewed variable neighborhood search (SVNS), introduced by DVC, is explained together with minor adjustments I implemented. This search is used in the final step of the metaheuristic algorithm. Also new alternative SVNSs are offered. The other subsections are dedicated to the phases that have to be executed prior to and within the SVNSs.

## 3.1 Structure of the algorithm

The main purpose behind a SVNS is just like an ordinary variable neighborhood search (VNS) to start of with an initial solution for a problem and attempt to improve this solution. Where the VNS only examines neighboring solutions of the best solution found so far, the SVNS also considers neighboring solutions of slightly worse solutions than the actual best found solution. This characteristic of the SVNS allows it to also investigate the search space further away from the initial solution and more extensively than the VNS does.

As stated above, before applying the SVNS, an initial solution is required. To reduce the total computation time it is preferred to start the SVNS with a good solution. The procedure from which I obtained the initial solution is described in section 3.2.

The SVNS itself basically exists out of two phases: 1)The shaking phase, in which a neighboring solution is found, and 2) the recentering phase, which evaluates whether the neighboring solution is better than the current best found solution or maybe good enough to serve as starting solution for the next iteration. Within the initialization phase as well as in the shaking phase a *Local Search* is applied to improve intermediate solutions. The *Local Search* algorithm and the *Local Search* moves used are described in section 3.4.

Due to the recentering phase in the SVNS proposed by Divsalar et al.(2013), the search tends to end up in an infinite loop. To strengthen this statement, it is useful to first discuss the methods used in the shaking phase more explicitly. Therefore, further explanations on this matter are in section 3.5. To avoid ending up in an infinite loop I slightly adjusted the SVNS by DVC and came up with two SVNS methods, these will be referred to as SVNS1 and SVNS2. Overviews of these SVNSs are presented in Algorithm 1 and Algorithm 2 respectively. The actual name of Algorithm 1 is SVNS1-V, where '-V' refers to the vertices shake that is used in the algorithm. The necessity of the additional letter is elucidated in Section 3.3.

For the SVNS1, the maximum number of iterations without improving the best found solution, *NoImprovementMax*, is set to 50 and the maximum percentage a solution is allowed to be lower than the prior solution to still be recentered is 0.03, *MaxPercentageWorse*. The value for *kMax* differs for every OPHS. The computation of this parameter is clarified in section 3.3. The values I used for these parameters are based on DVC who tested for parameter sensitivity in their paper.

For the SVNS2, the same values for *kMax* and *MaxPercentageWorse* are used as for SVNS1. Due to the different structure of SNVS2, it is unnecessary to have a evenly high value for *NoImprovementMax*, and is set to 3. Also further explanation on this follows after the clarification of the shaking phase in section 3.5.

**Algorithm 1** SVNS1-V

```
Input Initial Solution; Ranked list with Used Feasible Hotels Combinations;
    Kmax;
Initialize X, BestFound ← Initial Solution; NoImprovement ← 0;
    NoImprovementMax ← 50; MaxPercentageWorse ← 0.03;

    while NoImprovement < NoImprovementMax do
        K ← 1;
        while K ≤ Kmax do
            X' ← Vertices-Shake(X);
            if X' better than BestFound then
                BestFound ← X';
            end if
            X"← Hotels-Shake(X');
            Recenter or not?
            if X" better than BestFound then
                Recenter: X, BestFound ← X";
                NoImprovement ← 0; K ← 1;
            else
                NoImprovement+1; K+1;
                if Score(X") ≥ (1-MaxPercentageWorse) × Score(X) then
                    Recenter: X ← X";
                end if
            end if
        end while
    end while
    return BestFound;
```

**Algorithm 2** SVNS2-V

```
Input Initial Solution; Ranked list with Used Feasible Hotels Combinations;
  Kmax;
Initialize BestFound $\leftarrow$ Initial Solution; K $\leftarrow$ 1;
  for K $\leftarrow$ 1, Kmax do
    NoImprovement $\leftarrow$ 0;
    X $\leftarrow$ Initialize(K);
    while NoImprovement $<$ 3 do
      X' $\leftarrow$ Vertices-Shake(X);
      if X' better than BestFound then
        NoImprovement $\leftarrow$ 0;
        X, BestFound $\leftarrow$ X';
      else
        NoImprovement+1;
        if Score(X') $\geq 0.997 \times$ Score(X) then
          X $\leftarrow$ X';
        end if
      end if
    end while
  end for
  return BestFound;
```

## 3.2 Initialization phase

The initialization phase can roughly be divided into two steps. The first step focuses on determining the best feasible hotel combinations, being the hotels the tourist spends his nights. In the second step, feasible tours for the best hotel combinations are constructed.

Primarily, all feasible hotel combinations in the OPHS are determined, using the pruning rule. Next, a score is assigned to every possible hotel pair, containing the starting hotel $i$ and ending hotel $j$, for day $d = 1, ..., D$, by solving a sub-OP (planning a trip between hotel $i$ and $j$) heuristically with $T_d$ as the maximum length and all hotspots in the OPHS being considered for insertion for each trip. Since the tourist is possible to start and end a day trip at the same hotel, each hotel also forms a pair with itself. If $t_{ij} > T_d$, the score for the hotel pair is set to $-\infty$. Every sub-OP is solved by a simple *Greedy sub-OP Heuristic*, see Algorithm 3, in which a newly obtained trip after an iteration is assumed to be better when the total score of the trip has increased, or the score has remained equal, but the length of the trip has decreased. The local search methods used in the heuristic are described in section 3.4. All scores corresponding to the found solutions to the sub-OPs are stored in an $H \times H \times D$ matrix.

Now for every feasible hotel combination the *heuristic estimated score* (HES) is calculated by summing the previously obtained scores from every hotel pair that is present in the hotel combination on each day. The HES is only a rough estimation of the score of the final solution, since within the tour corresponding to the HES, the possibility exists that a hotspot is visited twice and therefore also counted twice for the HES. On the other hand the sub-OPs were solved heuristically and better sub-OPs might exist. So the optimal score of the is

**Algorithm 3** Greedy Sub-OP Heuristic

```
Input Start hotel $s$; End hotel $e$; Day;
Initialize Y ← infeasible Trip; Y' ← empty Trip with $s$ and $e$;
  while Y' is better than Y do
    Y ← Y';
    Insert(Y');
    Replacement(Y');
    Two-Opt(Y');
    Move-best(Y');
  end while
  return Y;
```

possible to be higher than the highest obained HES. All hotel combinations are ranked by their HES. Only a number of best hotel combinations based on the HES is used in the next step. Let this number be: 'the number of used hotel combinations' (NUFC).

Next, three strategies are applied to the selected set of hotel combinations to create feasible tours and finally also the initial solution for the SVNS.

1. *Local Search* is applied to a tour in which only the hotels are visited and no hotspots.
2. The sub-OP greedy heuristic is used to design trips, day by day, starting with the first day. *Local Search* is applied to improve the solution further.
3. The sub-OP greedy heuristic is used to design trips, day by day, starting with the last day. *Local Search* is applied to improve the solution further.

In strategies 2 and 3 the hotspots considered for insertion in the sub-OP greedy heuristic are only those which are not yet present in previous designed day trips. Therefore the obtained tour are impossible to contain hotspots multiple times. The strategy which results in the tour with the highest score is stored together with the tour itself. Next, the selected set of hotel combinations is again ranked according to the score of the best found tour. This ranked list existing out of NUFC hotel combinations is used again in the shaking phase described in section 3.3. The tour found with the highest score is set as the initial solution to start the SVNS.

## 3.3 Shaking Phase

The shaking phase proposed by DVC contains a Vertices-Shake and a Hotels-Shake. Both "shakes" are present in SVNS1. The SVNS2 contains the initialize method instead of the Hotels-shake. I also introduced a trips-shake method, a method to find different tours to attempt to improve the solution. All methods are described in this section.

The Vertices-Shake does not guarantee an improvement of the prior solution and might even lead to a slight decrease of the score. However, it possibly moves away from the prior solution, and gets closer to a better or even the optimal solution. The Vertices-Shake uses two strategies to adjust the current solution.

1. Remove the first halves of the hotspots for each day trip and improve the remaining solution by applying *Local Search*.

2. Remove the second halves of the hotspots for each day trip and improve the remaining solution by applying *Local Search*.

The SVNS then continues with the best solution obtained from one of the two strategies. If this solution appears to be better than the best found solution so far, the best found solution is immediately replaced by the new best found solution.

In the Hotels-shake, the current hotels are replaced. This method is only used in SVNS1. The hotel combination of the current solution is possibly not the hotel combination corresponding with the optimal tour. Before the hotel shake can be implemented a new hotel combination has to be specified. In the SVNS this new hotel combination is indicated with an integer $K$ corresponding with the rank of the new hotel combination in the last ranked list obtained in the initialization phase. Once the new hotel combination is specified, the hotel of the current solution are replaced by those of the new hotel combination at exactly the same position in the tour. This is most likely to end up in an infeasible tour. Therefore hotspots are removed from the tour one by one based on the least ratio score over time saved by removal. Hotspots are removed until the tour is feasible. Then again *Local Search* is applied to further improve the solution. The total number of alternative hotel combinations that are considered in the hotel shake is limited by $Kmax$. Following the test results on parameter sensitivity by DVC. (2013), I set NUFC to 250 and $Kmax$ to be $\lceil 0.25 \times NUFC \rceil = 63$. If the total number of feasible combinations (TNFC) for the OPHS is lower than NUFC, NUFC is set equal to TNFC and $Kmax = min\{TNFC, 63\}$.

Since the Hotels-Shake actually forces a hotel combination into a tour, which was designed for another hotel combination, it appears inconvenient to find its best tour starting from such an odd solution. Therefore in the SVNS2, the initialize method is used which designs a starting solution for the $K$th ranked hotel combination using the best out of the three strategies offered in the initialization phase. The value for $Kmax$ for the SVNS2 is still computed in the same way as for the SVNS1.

The last shake method I introduced is the Trips-Shake and can be used as a replacement of or in addition to the Vertices-Shake. The Trips-Shake designs new trips, day by day, starting with the first day, using the sub-OP Greedy Heuristic. In the sub-OP Greedy Heuristic, all currently inserted hotspots are excluded from insertion, also the hotspots included in trip that is going to be replaced. After having replaced a trip *Local Search* is applied to improve the solution. Replacing the Vertices-Shake by the Trips-Shake results in two new SVNSs, named SVNS1-T and SVNS2-T. For explicitness, the SVNSs in which the Vertices-Shake is used and no Trips-Shake are now named SVNS1-V and SVNS2-V.

Obviously, also the possibility exists to combine the Vertices-Shake and the Trips-Shake. In that case, the best solution evolving from one of the two 'Shakes' is used to continue the SVNS. This again result in two extra variants on the original SVNS. These are named SVNS1-VT and SVNS2-VT.

## 3.4 Local Search

In this section the Local Search algorithm is described together with the Local Search moves used in the Local Search and in the sub-OP Greedy Heuristic.

**Algorithm 4** Local Search

```
  Local Search Methods: Insert(.)=Local[1](.); Move-Best(.)=Local[2](.);
  Two-Opt(.)=Local[3](.); Swap-Trips(.)=Local[4](.);
  Extract-Insert(.)=Local[5](.); Extract2-Insert(.)=Local[6](.);
  Extract5-Insert(.)=Local[7](.); Extract-Move-Insert(.)=Local[8](.);
Input Initial Solution;
Initialize X ← Infeasible Solution; Level ← 1;
  while Level $\leq$ 8 do
    X' ← Local[Level](X);
    if X' better than X then
      X' ← X; Level ← 1;
    else
      Level+1;
    end if
  end while
  return BestFound;
```

An overview of the Local Search algorithm is presented in Algorithm 4. The algorithm is used to improve feasible predefined solutions, The Local Search moves are in general adapted from DVC, with two moves being small adjusted. An important characteristic of the Local Search is that it stops as soon as the solution cannot be improved anymore by one of the local search moves, and also never worsens the predefined solution. Besides, the solution always remains feasible after each move. Next each Local Search move is described briefly.

1. *Insert*: For every hotspot, that is currently excluded from the tour, the optimal position of insertion is determined, meaning that for inserting vertex $i$ at position $v$ in the tour, the tour remains feasible and the increase in time length of the tour is minimized. The vertex with the maximum ratio of score over increase in time is inserted.

2. *Move-best*: For each included hotspot, it is checked that whether moving the hotspot to a different position in the tour leads to a decrease in time. The movement which leads to the highest decrease in time is executed. Hence, this move is slightly different from the way *Move-best* it is applied by DVC.

3. *Two-opt*: For each hotspot, starting with the first hotspot in the tour, it is examined whether inverting the hotspot with any other hotspot in the same trip results in a decrease of the travel time. If any inversion results in a decrease of the travel time the order between the vertices for which an inversion leads to the highest time save are reversed. If an inversion takes place, the procedure starts again from the first hotspot of the tour in which the last inversion took place.

4. *Swap-trips*: It is very similar to the *Two-opt* move, but now it is examined whether inverting two hotspots from two different trips reduces the total

travel time, starting from the first hotspot that is visited in the tour. When swapping the two hotspots, both trips should remain feasible and the total travel time must decrease. If there appear to be multiple options to swap the current hotspot with another, the swap resulting in the largest decrease in total travel time is made. If a swap occurred, the examination starts from the first hotspot in the tour again

5. *Extract-Insert*: Starting from the first hotspot in the tour, a hotspot is extracted from the tour, then *Insert* is applied until it is not possible any longer. The excluded hotspot is not considered for insertion. The move is only executed when the new solution is better than the previous one. If a solution with a higher score is found, the first vertex of the tour is up for the same procedure again, otherwise it moves on to the next hotspot in the tour.

6. *Extract2-Insert*: Very similar to the previous move, but now two subsequent hotspots are extracted from the tour and both excluded vertices are not considered for insertion again.

7. *Extract5-Insert*: Again similar to the two moves above, but this time five subsequent hotspots are extracted. Note, this move is only applied when the trip exists out of at least five hotspots.

8. *Extract-Move-Insert*: The first hotspot visited in the tour is extracted. Then *Move-best* is applied, and at last *Insert*, where only excluded hotspots with a higher score are considered for insertion. Only if a new insertion takes place, the entire procedure is applied and the same process will start from the first hotspot on again. Otherwise it moves on the next hotspot in the tour. Also this move slightly diverges from the *Extract-Move-Insert* described by Divsalar et al. (2013).

9. *Replacement*: It first tries to apply the *Insert* move. If it is not successful: for each non-included hotspot, the best position of insertion is determined. The hotspot is included the trip. The solution has now become infeasible. Only if extracting one hotspot with a lower score than the inserted hotspot results in a feasible trip again, the respective hotspots are inserted and extracted. Nonetheless, it moves on to the next non-included vertex to repeat the same procedure.

## 3.5 Recentering Phase

This section first clarifies the decision to diverge from the SVNS used in DVC. Next the section describes the recentering phase in the SVNS1 as well as the one in the SVNS2, together with their advantages and disadvantages.

The SVNS proposed by DVC is very similar to SVNS1. The main difference between the two is that in the SVNS proposed by DVC, $K$ is set back to 1 after any recentering. Once a slightly worse solution is found, and is recentered, in the next iteration, the hotel combination of this centered solution is replaced by the best found hotel combination according to the initialization phase. It is probable that the next new solution is better than the currently centered solution. Again $K$ is set to 1, as soon as a slightly worse solution is centered again, the same process is likely to happen again, which results in the algorithm getting stuck in the inner while-loop. One remark, unlike the algorithm presented in Algorithm 1, the SVNS proposed by DVC, does not recenter when a new solution after an iteration is found with the same score the new solution is not recentered. This prevents from recentering the same solution over and over again. However, this does not prevent the algorithm from ending up in an infinite loop as described earlier.

Within the SVNS1, the solution obtained at the end of an iteration is recentered if the solution is at most 0.3 per cent worse than the currently centered solution. $K$ is only lowered when the best found solution has increased, or the inner while-loop has been finished, but the number of iterations without improvement has not yet exceeded its maximum. Since the hotel combinations change very quickly, it appears that the neighborhood of an individual hotel combination is explored insufficiently. Definitely, if one recalls the fact that the hotel combination is forced into a tour that was basically designed for a different hotel c

To make sure that the neighborhood for every hotel combination that is considered is investigated sufficiently, I introduced the SVNS2. Instead of using the Hotels-Shake to evaluate different hotel combinations. The initialization method is used. The neighborhood search for each hotel combination with at most rank $Kmax$ after the initialization phase is similarly explored. The tour related to the hotel combination is 'shaken' at least three times, *NoImprovementMax*. If the tour turns out better than the best found tour so far, the area around the new best found tour is further explored. The value for *NoImprovementMax* is kept so low to reduce the total computation time and because increasing it to did not appear to improve the best found solution.

## 4 Experiments and results

This section is dedicated to the performances of the metaheuristic algorithm with multiple variations of the SVNS, described in this paper. They are compared to performances of the metaheuristic algorithm obtained by DVC. The test instances that are used the same as the instances solved in DVC. A brief description of the design of the test instances and how they were constructed is given in section 4.1.

### 4.1 Test Instances

Since there were no OPHS test instances available for DVC, they designed four different sets of benchmark instances to evaluate the performance of their metaheuristic algorithm. The first three sets of OPHS instances were derived from existing OP instances used in Tsiligiridis (1984) and Chao et al. (1996), by an ingenious technique . The major advantage of this technique is that the optimal tour and the corresponding score for the OPHS instance is exactly the same as the optimal tour for the OP instance. One consequence of the technique is that the hotel combination in the optimal tour always contains D+1 different hotels, in other words the tourist will, when doing the optimal tour, spend every night in a different hotel.

For SET 1: in total 105 OPHS instances were designed. Where 35 test instances, for which H=3 and D=2, 35 test instances for which H=4 and D=3, and another 35 test instances for which H=5 and D=4.

The last 70 test instances are used construct more complex instances in SET 2. To each instance 3 extra hotels are added, which do not influence the optimal tour. SET 2 consequently contains 35 test instances where H=7 and D=3 and 35 test instances where H=8 and D=4.

For SET 3, even more complex instances are designed. 22 instances where H=12 and D=4 and 22 instances where H=14 and D=5. These instances are designed similarly as in SET 1 and SET 2.

SET 4 contains 6 test instances where H=5 and D=2 and 6 test instances where H=5 and D=3. For these test instances, the optimal solution is not known in advance. The instance does evolve from an OP problem, but the hotels that are added, are located randomly. To get a benchmark for these instances, a sub-OP is solved up to optimality for each possible hotel pair. Like in the initialization phase the HES is computed for each possible hotel combination. The retrieved tour possibly includes hotspots that are visited twice. To obtain an upper bound for the instances, the highest HES is chosen. If all hotspots in the tour corresponding with the highest HES are visited only once, this tour then obviously serves as optimal solution of the OPHS. However, if at least one hotspot is visited twice in the tour, a lower bound to the OPHS is obtained by removing the duplicated hotspots. The optimal solution for the 6 test instances where D=2 are found without an exception. On the other hand, no optimal solution was found for any of the instances where D=3.

Table 1: SET1: H=3 and D=2

| S - T | OptVal | DVC | Init. | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|---|
| 30-65 | 240 | **240** | 235 | 235 | 235 | 235 | 235 | 235 | 235 |
| 30-70 | 260 | **260** | **260** | **260** | **260** | **260** | **260** | **260** | **260** |
| 30-73 | 265 | 245 | 245 | 250 | 250 | **265** | **265** | **265** | **265** |
| 30-75 | 270 | **270** | **270** | **270** | **270** | **270** | **270** | **270** | **270** |
| 30-80 | 280 | 270 | 270 | 270 | 270 | 270 | 270 | 270 | 270 |
| 30-85 | 285 | 280 | 280 | 280 | 280 | 280 | 280 | 280 | 280 |
| 30-65 | 610 | **610** | **610** | **610** | **610** | **610** | **610** | **610** | **610** |
| 30-75 | 670 | 650 | 650 | 650 | 650 | **670** | **670** | **670** | **670** |
| 30-80 | 710 | 690 | 680 | 680 | 680 | 700 | 700 | 700 | 700 |
| 30-85 | 740 | **740** | 720 | 720 | 720 | 720 | 720 | 720 | 720 |
| 30-90 | 770 | **770** | **770** | **770** | **770** | **770** | **770** | **770** | **770** |
| 30-95 | 790 | 780 | 750 | 760 | 760 | 750 | 750 | 760 | 760 |
| 30-100 | 800 | 770 | 760 | 760 | 760 | 760 | 760 | 760 | **800** |
| 30-105 | 800 | **800** | 790 | **800** | **800** | 790 | 790 | **800** | **800** |
| 62-45 | 816 | **816** | 810 | **816** | **816** | 810 | 810 | **816** | **816** |
| 62-50 | 900 | 876 | 852 | 858 | 858 | 870 | 870 | 876 | 876 |
| 62-55 | 984 | 972 | 942 | 972 | 972 | 942 | 942 | 972 | 972 |
| 62-60 | 1062 | 1050 | 1044 | 1044 | 1044 | **1116** | **1116** | **1116** | **1116** |
| 62-65 | 1116 | **1116** | 1068 | **1116** | **1116** | 1068 | 1068 | **1116** | **1116** |
| 62-70 | 1188 | 1170 | 1164 | 1164 | 1164 | 1164 | 1164 | 1170 | 1170 |
| 62-75 | 1236 | **1236** | 1212 | 1230 | 1230 | 1212 | 1212 | 1230 | 1230 |
| 62-80 | 1284 | **1284** | 1272 | 1272 | 1272 | 1272 | 1272 | 1272 | 1272 |
| 64-40 | 575 | 570 | 550 | 570 | 570 | 550 | 550 | 570 | 570 |
| 64-45 | 650 | 645 | 625 | 645 | 645 | 625 | 625 | 645 | 645 |
| 64-50 | 730 | 715 | 690 | 700 | 700 | 690 | 690 | 700 | 700 |
| 64-55 | 825 | 805 | 805 | 805 | 805 | 805 | 805 | 805 | 805 |
| 64-60 | 915 | 860 | 840 | 890 | 890 | 840 | 840 | 890 | 890 |
| 64-125 | 1670 | 1665 | 1640 | 1640 | 1640 | 1640 | 1640 | 1640 | 1640 |
| 64-130 | 1680 | **1680** | 1675 | 1675 | 1675 | 1675 | 1675 | 1675 | 1675 |
| 99-30 | 173 | **173** | **173** | **173** | **173** | **173** | **173** | **173** | **173** |
| 99-35 | 241 | **241** | **241** | **241** | **241** | **241** | **241** | **241** | **241** |
| 99-40 | 299 | **299** | **299** | **299** | **299** | **299** | **299** | **299** | **299** |
| 99-45 | 367 | **367** | **367** | **367** | **367** | **367** | **367** | **367** | **367** |
| 101-50 | 181 | **181** | **181** | **181** | **181** | **181** | **181** | **181** | **181** |
| 101-60 | 243 | **243** | **243** | **243** | **243** | **243** | **243** | **243** | **243** |

## 4.2 Results

In this section, the solutions found by the various metaheuristic algorithms for the different SETs of OPHS instances are presented. The outcomes are compared to the outcomes produced by DVC. The ranks of the hotel combinations of the best found solution by each SVNS described in this paper, are discussed. Next to that the computation times are discussed and again compared to DVC. Due to the size of the tables which show the obtained solutions, most of them are only presented in the appendix.

All OPHS instances that are contained by SET1, SET2 and SET4 were solved by the metaheuristic algorithm using SVNS1-V, SVNS2-V, SVNS1-T, SVNS2-T, SVNS1-VT and SVNS2-VT. The instances in SET3 were only solved using the SVNS1-V, SVNS2-V, SVNS1-T and SVNS2-T, due to high computation times when using the SVNS1-VT and SVNS2-VT.

Table 1 shows the highest score found by each solving method for each OPHS instance in SET1, for which H=3 and D=2. The most left column denotes the number of hotspots present in the traveling area and the maximum length of the Tour. The second column shows the optimal value of each particular instance. The third column presents the highest scores that were found by DVC. Every value on the right hand of line were obtained in this research. The most left column on the right side shows the highest score obtained after the initialization phase. The remaining columns on the right side denote the best scores found by the solving methods with a varying SVNS, When the score is in bold it means that the particular solving method was able to find the optimal solution of the OPHS. When the score is underlined, it means that even though the optimal solution was not found, the solving method did perform the best for that particular instance. The scores that were found for each solving method for the other instances in SET1, SET2 and SET3 are present in Appendix A.

For every instance in SET1, SET2 and SET3 the optimal value is known. Therefore it is easy to compute a measure for the results, named the optimality gap. The optimality gap is defined as
*(OptimalResult-BestFoundResult)/OptimalResult*. The optimality gap is computed for each instance for every solving method. The averages of these optimality gaps per instance sort can be found in table 3.

As described earlier, the optimal results for the instances in SET4 are not all known. The instances without a known optimal value have a an upper bound and a lower bound. The results of the solving method for the instances in SET4 are presented in table 2. To compare the solving methods, the results found by DVC are used as a benchmark. The gap for each solving method for each instance in SET4 is calculated by *(DVCResult-BestFoundResult)/DVCResult*. The average gaps for each solving method can also be found in table 3. The 'Best' column in table 3 represents the gaps when for every instance the best result of the six SVNSs is chosen.

Table 2: SET4 - H=5 and D=2/3

| S - T - D | OptVal | LB | UB | DVC | Init. | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 98-20-2 | 247 | | | 247 | 240 | 240 | 240 | 240 | 240 | 240 | 240 |
| 98-30-2 | 385 | | | 385 | **385** | **385** | **385** | **385** | **385** | **385** | **385** |
| 100-35-2 | 157 | | | 151 | 151 | 151 | 151 | 151 | 151 | 151 | 151 |
| 100-40-2 | 210 | | | 210 | **210** | **210** | **210** | **210** | **210** | **210** | **210** |
| 100-45-2 | 266 | | | 266 | 254 | 254 | 254 | 254 | 254 | 254 | 254 |
| 98-20-3 | | 357 | 376 | 368 | 368 | 368 | 368 | 368 | 368 | 368 | 368 |
| 98-25-3 | | 495 | 568 | 524 | 524 | 524 | 524 | 524 | 524 | 524 | 524 |
| 100-30-3 | | 230 | 380 | 324 | 300 | 300 | 300 | 300 | 300 | 300 | 300 |
| 100-40-3 | | 299 | 493 | 383 | 366 | 383 | 383 | 387 | 387 | 383 | 383 |
| 100-45-3 | | 356 | 579 | 442 | 420 | 430 | 425 | 425 | 425 | 425 | 427 |

Table 3: The Average Gap (AG) for each test instance

| | H-D | DVC | Init. | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT | Best |
|---|---|---|---|---|---|---|---|---|---|---|
| SET1 | 3-2 | 1.22 | 2.61 | 1.71 | 1.71 | 1.91 | 1.91 | 1.15 | 1.00 | 1.00 |
| | 4-3 | 0.93 | 1.90 | 1.42 | 1.31 | 1.90 | 1.63 | 1.42 | 1.15 | 1.15 |
| | 5-4 | 0.92 | 1.88 | 1.56 | 1.50 | 1.75 | 1.75 | 1.47 | 1.41 | 1.23 |
| SET2 | 7-3 | 0.98 | 1.56 | 1.17 | 1.15 | 1.56 | 1.37 | 1.17 | 0.98 | 0.93 |
| | 8-4 | 1.22 | 1.89 | 1.48 | 1.58 | 1.68 | 1.74 | 1.39 | 1.48 | 1.24 |
| SET3 | 12-4 | 2.61 | 4.32 | 3.10 | 3.43 | 4.15 | 4.15 | - | - | 2.84 |
| | 14-5 | 3.58 | 4.49 | 4.09 | 3.82 | 4.49 | 4.49 | - | - | 3.69 |
| SET4* | 5-2/3 | 0.00 | 2.42 | 1.75 | 1.86 | 1.76 | 1.76 | 1.86 | 1.81 | 1.64 |
| Overall AG** | | 1.46 | 2.46 | 1.90 | 1.89 | 1.86 | 1.79 | 1.32*** | 1.20*** | 1.54 |

*Not all optimal values in SET4, therefore the value presented is the average gap to the solutions found by DVC
**The Overall AG only considers the instances in SET1, SET2 and SET3
***The Overall AG for this value is based only on the instances in SET1 and SET2

Also the computation times for solving the OPHS instances are often discussed by DVC. The average computation time for each solving method for each test instance is presented in table 4 and can be compared to the computations times stated in DVC.

Table 4: The Average Computation Time (ACPU) for each instance set

| | H-D | DVC | Init. | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|---|
| SET1 | 3-2 | 0.18 | 0.20 | 2.21 | 0.45 | 1.75 | 0.32 | 4.87 | 0.79 |
| | 4-3 | 0.21 | 0.63 | 3.08 | 1.31 | 2.74 | 1.13 | 5.77 | 2.93 |
| | 5-4 | 0.65 | 3.76 | 5.94 | 6.24 | 5.64 | 5.59 | 8.71 | 10.11 |
| SET2 | 7-3 | 0.34 | 2.06 | 6.05 | 4.55 | 5.64 | 3.69 | 10.18 | 8.15 |
| | 8-4 | 1.39 | 9.02 | 11.22 | 11.86 | 11.16 | 10.97 | 14.19 | 16.55 |
| SET3 | 12-4 | 8.20 | 56.68 | 75.72 | 79.38 | 74.96 | 71.40 | - | - |
| | 14-5 | 6.77 | 50.05 | 64.67 | 73.48 | 71.14 | 70.00 | - | - |
| SET4 | 3-2/3 | 0.24 | 0.47 | 2.57 | 0.91 | 2.62 | 0.89 | 6.55 | 1.96 |
| Overall ACPU | | 1.95 | 13.23 | 18.66 | 19.26 | 11.45 | 10.50 | 8.74* | 7.71* |

*The Overall ACPU is only based on the computation times of the instances in SET1, SET2 and SET4

Table 5: The number of Best Found solutions without the highest ranked hotel combination after the initialization phase

| | H - D | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|
| SET1 | 3-2 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 4-3 | 1 | 1 | 0 | 1 | 1 | 2 |
| | 5-4 | 1 | 2 | 0 | 0 | 1 | 2 |
| SET2 | 7-3 | 1 | 2 | 0 | 1 | 1 | 3 |
| | 8-4 | 3 | 3 | 1 | 1 | 3 | 4 |
| SET3 | 12-4 | 8 | 6 | 2 | 3 | - | - |
| | 14-5 | 2 | 7 | 0 | 1 | - | - |
| SET4 | 5-2/3 | 1 | 0 | 0 | 0 | 0 | 0 |
| Total | | 17 | 21 | 3 | 7 | 6 | 12 |

The reason to introduce the SVNS2 was to make sure, also the neighborhood of solutions with a lower ranked hotel combination are investigated sufficient. Table 5 shows the number of times the highest ranked hotel combination is *not* used in the best found solution for each subset of OPHS instances. The ranks of the actual hotel combinations used in the best found solutions can be found in the tables in Apppendix B.

## 5 Conclusion

The varying solving methods that I introduced resulted in different best found solution. Only for the instances of SET1 where H=3 and D=2, two individual solving methods, SVNS-1VT and SVNS-2VT, performed better than the SVNS by DVC. Jointly the solution methods I introduced also performed better for the instances in SET2 where H=7 and D=3. Also in several individual instances the SVNS by DVC is outperformed by one of other solving. In general, however DVC were able to find better solutions to the OPHS instances which is shown by table 3. Also the efficiency of the SVNS by DVC is impressive when comparing the computation times I obtained.

About the differences between the SVNSs introduced in this paper. Overall the SVNSs only using the Vertices-Shake outperform the SVNSs only using the Trips-Shake. When used together however the average optimality gap is significantly decreased. The disadvantage of using both 'Shakes' is a rise in the average computation time.

The differences between the SVNS1 and the SVNS2 are smaller than I expected. Table 5 shows that the SVNS2 slightly more often leads to a best found solution with a hotel combination that was ranked second or lower after the initialization phase, the difference in number times however is negligible. Both methods do regularly end up with a different best solution and can therefore complement each other.

## References

[1] Chao, I.-M., Golden, B.L., Wasil, E.A., 1996. Theory and methodology-a fast and effective heuristic for the orienteering problem. European Journal of Operational Research 88 (3), 475-489.

[2] Divsalar, A., Vansteenwegen, P., Cattrysse, D. 2013, A variable neighboorhood search method for the orienteering problem with hotel selection. Int. J. Production Economics 145, 150-160.

[3] Tsiligirides, T., 1984. Heuristic methods applied to orienteering. Journal of the Operational Research Society 35 (9), 797-809.

[4] Vansteenwegen, P., Souffriau, W., Berghe, G., Oudheusen, D., 2009. Metaheuristics for tourist trip planning. Lecture Notes in Economics and Mathematical Systems 624, 15-31.

## A Scores of the Best Found Solution per Instance per SVNS

These tables show the highest score found by each solving method for each OPHS instance. The most left column denotes the number of hotspots present in the traveling area and the maximum length of the Tour. The second column shows the optimal value of each particular instance. The third column presents the highest scores that were found by DVC. Every value on the right hand of line were obtained in this research. The most left column on the right side shows the highest score obtained after the initialization phase. The remaining columns on the right denote the best scores found by the solving methods with a varying SVNS, When the score is in bold it means that the particular SVNS was able to find the optimal solution of the OPHS. When the score is underlined, it means that even though the optimal solution was not found, the solving method did perform the best for that particular instance.

Table 6: SET1 - H=3 and D=2

| S-T | OptVal | DVC | Init. | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|---|
| 30-65 | 240 | **240** | $\underline{235}$ | $\underline{235}$ | $\underline{235}$ | $\underline{235}$ | $\underline{235}$ | $\underline{235}$ | $\underline{235}$ |
| 30-70 | 260 | **260** | **260** | **260** | **260** | **260** | **260** | **260** | **260** |
| 30-73 | 265 | 245 | 245 | 250 | 250 | **265** | **265** | **265** | **265** |
| 30-75 | 270 | **270** | **270** | **270** | **270** | **270** | **270** | **270** | **270** |
| 30-80 | 280 | 270 | $\underline{270}$ | $\underline{270}$ | $\underline{270}$ | $\underline{270}$ | $\underline{270}$ | $\underline{270}$ | $\underline{270}$ |
| 30-85 | 285 | 280 | $\underline{280}$ | $\underline{280}$ | $\underline{280}$ | $\underline{280}$ | $\underline{280}$ | $\underline{280}$ | $\underline{280}$ |
| 30-65 | 610 | **610** | **610** | **610** | **610** | **610** | **610** | **610** | **610** |
| 30-75 | 670 | 650 | 650 | 650 | 650 | **670** | **670** | **670** | **670** |
| 30-80 | 710 | 690 | 680 | 680 | 680 | $\underline{700}$ | $\underline{700}$ | $\underline{700}$ | $\underline{700}$ |
| 30-85 | 740 | **740** | $\underline{720}$ | $\underline{720}$ | $\underline{720}$ | $\underline{720}$ | $\underline{720}$ | $\underline{720}$ | $\underline{720}$ |
| 30-90 | 770 | **770** | **770** | **770** | **770** | **770** | **770** | **770** | **770** |
| 30-95 | 790 | 780 | 750 | $\underline{760}$ | $\underline{760}$ | 750 | 750 | $\underline{760}$ | $\underline{760}$ |
| 30-100 | 800 | 770 | 760 | 760 | 760 | 760 | 760 | 760 | **800** |
| 30-105 | 800 | **800** | 790 | **800** | **800** | 790 | 790 | **800** | **800** |
| 62-45 | 816 | **816** | 810 | **816** | **816** | 810 | 810 | **816** | **816** |
| 62-50 | 900 | 876 | 852 | 858 | 858 | 870 | 870 | $\underline{876}$ | $\underline{876}$ |
| 62-55 | 984 | 972 | 942 | 972 | 972 | 942 | 942 | $\underline{972}$ | $\underline{972}$ |
| 62-60 | 1062 | 1050 | 1044 | 1044 | 1044 | **1116** | **1116** | **1116** | **1116** |
| 62-65 | 1116 | **1116** | 1068 | **1116** | **1116** | 1068 | 1068 | **1116** | **1116** |
| 62-70 | 1188 | 1170 | 1164 | 1164 | 1164 | 1164 | 1164 | $\underline{1170}$ | $\underline{1170}$ |
| 62-75 | 1236 | **1236** | 1212 | $\underline{1230}$ | $\underline{1230}$ | 1212 | 1212 | $\underline{1230}$ | $\underline{1230}$ |
| 62-80 | 1284 | **1284** | $\underline{1272}$ | $\underline{1272}$ | $\underline{1272}$ | $\underline{1272}$ | $\underline{1272}$ | $\underline{1272}$ | $\underline{1272}$ |
| 64-40 | 575 | 570 | 550 | $\underline{570}$ | $\underline{570}$ | 550 | 550 | $\underline{570}$ | $\underline{570}$ |
| 64-45 | 650 | 645 | 625 | $\underline{645}$ | $\underline{645}$ | 625 | 625 | $\underline{645}$ | $\underline{645}$ |
| 64-50 | 730 | 715 | 690 | $\underline{700}$ | $\underline{700}$ | 690 | 690 | $\underline{700}$ | $\underline{700}$ |
| 64-55 | 825 | 805 | $\underline{805}$ | $\underline{805}$ | $\underline{805}$ | $\underline{805}$ | $\underline{805}$ | $\underline{805}$ | $\underline{805}$ |
| 64-60 | 915 | 860 | 840 | $\underline{890}$ | $\underline{890}$ | 840 | 840 | $\underline{890}$ | $\underline{890}$ |
| 64-125 | 1670 | 1665 | $\underline{1640}$ | $\underline{1640}$ | $\underline{1640}$ | $\underline{1640}$ | $\underline{1640}$ | $\underline{1640}$ | $\underline{1640}$ |
| 64-130 | 1680 | **1680** | $\underline{1675}$ | $\underline{1675}$ | $\underline{1675}$ | $\underline{1675}$ | $\underline{1675}$ | $\underline{1675}$ | $\underline{1675}$ |
| 99-30 | 173 | **173** | **173** | **173** | **173** | **173** | **173** | **173** | **173** |
| 99-35 | 241 | **241** | **241** | **241** | **241** | **241** | **241** | **241** | **241** |
| 99-40 | 299 | **299** | **299** | **299** | **299** | **299** | **299** | **299** | **299** |
| 99-45 | 367 | **367** | **367** | **367** | **367** | **367** | **367** | **367** | **367** |
| 101-50 | 181 | **181** | **181** | **181** | **181** | **181** | **181** | **181** | **181** |
| 101-60 | 243 | **243** | **243** | **243** | **243** | **243** | **243** | **243** | **243** |

Table 7: SET1 - H=4 and D=3

| S-T | OptVal | DVC | Init. | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|---|
| 30-65 | 240 | **240** | **240** | **240** | **240** | **240** | **240** | **240** | **240** |
| 30-70 | 260 | **260** | **260** | **260** | **260** | **260** | **260** | **260** | **260** |
| 30-73 | 265 | **265** | **265** | **265** | **265** | **265** | **265** | **265** | **265** |
| 30-75 | 270 | **270** | 260 | 260 | 260 | 260 | 260 | 260 | 260 |
| 30-80 | 280 | **280** | 275 | **280** | **280** | 275 | 275 | **280** | **280** |
| 30-85 | 285 | **285** | 280 | 280 | 280 | 280 | 280 | 280 | 280 |
| 30-65 | 610 | **610** | **610** | **610** | **610** | **610** | **610** | **610** | **610** |
| 30-75 | 670 | 650 | 650 | 650 | 650 | 650 | 650 | 650 | 650 |
| 30-80 | 710 | **710** | **710** | **710** | **710** | **710** | **710** | **710** | **710** |
| 30-85 | 740 | **740** | 670 | 680 | 700 | 670 | **740** | 680 | **740** |
| 30-90 | 770 | 730 | 730 | 730 | 730 | 730 | 730 | 730 | 730 |
| 30-95 | 790 | **790** | **790** | **790** | **790** | **790** | **790** | **790** | **790** |
| 30-100 | 800 | **800** | **800** | **800** | **800** | **800** | **800** | **800** | **800** |
| 30-105 | 800 | 790 | 780 | 790 | 790 | 780 | 780 | 790 | 790 |
| 62-45 | 816 | **816** | 804 | 804 | 804 | 804 | 804 | 804 | 804 |
| 62-50 | 900 | 876 | 870 | 870 | 870 | 870 | 870 | 870 | 870 |
| 62-55 | 984 | 954 | 948 | 954 | 954 | 948 | 948 | 954 | 954 |
| 62-60 | 1062 | 1038 | 996 | 1026 | 1026 | 996 | 996 | 1026 | 1026 |
| 62-65 | 1116 | 1092 | 1080 | 1080 | 1092 | 1080 | 1080 | 1080 | 1098 |
| 62-70 | 1188 | 1170 | 1158 | 1170 | 1170 | 1158 | 1158 | 1170 | 1170 |
| 62-75 | 1236 | 1212 | 1200 | 1200 | 1200 | 1200 | 1200 | 1200 | 1200 |
| 62-80 | 1284 | 1260 | 1260 | 1260 | 1260 | 1260 | 1260 | 1260 | 1260 |
| 64-40 | 575 | 570 | 570 | 570 | 570 | 570 | 570 | 570 | 570 |
| 64-45 | 650 | 645 | 645 | 645 | 645 | 645 | 645 | 645 | 645 |
| 64-50 | 730 | 715 | 690 | 715 | 715 | 690 | 690 | 715 | 715 |
| 64-55 | 825 | 805 | 805 | **825** | **825** | 805 | 805 | **825** | **825** |
| 64-60 | 915 | 910 | 890 | 910 | 910 | 890 | 890 | 910 | 910 |
| 64-125 | 1670 | **1670** | 1645 | 1645 | 1645 | 1645 | 1645 | 1645 | 1645 |
| 64-130 | 1680 | 1665 | 1675 | 1675 | 1675 | 1675 | 1675 | 1675 | 1675 |
| 99-30 | 173 | **173** | **173** | **173** | **173** | **173** | **173** | **173** | **173** |
| 99-35 | 241 | **241** | **241** | **241** | **241** | **241** | **241** | **241** | **241** |
| 99-40 | 299 | **299** | **299** | **299** | **299** | **299** | **299** | **299** | **299** |
| 99-45 | 367 | **367** | **367** | **367** | **367** | **367** | **367** | **367** | **367** |
| 101-50 | 181 | **181** | **181** | **181** | **181** | **181** | **181** | **181** | **181** |
| 101-60 | 243 | **243** | **243** | **243** | **243** | **243** | **243** | **243** | **243** |

Table 8: SET1 - H=5 and D=4

| S-T | OptVal | DVC | Init. | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|---|
| 30-65 | 240 | **240** | **240** | **240** | **240** | **240** | **240** | **240** | **240** |
| 30-70 | 260 | **260** | **260** | **260** | **260** | **260** | **260** | **260** | **260** |
| 30-73 | 265 | **265** | **265** | **265** | **265** | **265** | **265** | **265** | **265** |
| 30-75 | 270 | **270** | **270** | **270** | **270** | **270** | **270** | **270** | **270** |
| 30-80 | 280 | 275 | $\underline{270}$ | $\underline{270}$ | $\underline{270}$ | $\underline{270}$ | $\underline{270}$ | $\underline{270}$ | $\underline{270}$ |
| 30-85 | 285 | **285** | **285** | **285** | **285** | **285** | **285** | **285** | **285** |
| 30-65 | 610 | **610** | **610** | **610** | **610** | **610** | **610** | **610** | **610** |
| 30-75 | 670 | 650 | $\underline{650}$ | $\underline{650}$ | $\underline{650}$ | $\underline{650}$ | $\underline{650}$ | $\underline{650}$ | $\underline{650}$ |
| 30-80 | 710 | **710** | **710** | **710** | **710** | **710** | **710** | **710** | **710** |
| 30-85 | 740 | **740** | $\underline{700}$ | $\underline{700}$ | $\underline{700}$ | $\underline{700}$ | $\underline{700}$ | $\underline{700}$ | $\underline{700}$ |
| 30-90 | 770 | **770** | $\underline{720}$ | $\underline{720}$ | $\underline{720}$ | $\underline{720}$ | $\underline{720}$ | $\underline{720}$ | $\underline{720}$ |
| 30-95 | 790 | **790** | 750 | 750 | **790** | 750 | 750 | 750 | **790** |
| 30-100 | 800 | 760 | 750 | **800** | 750 | 750 | 750 | **800** | 750 |
| 30-105 | 800 | **800** | 770 | 770 | $\underline{790}$ | 770 | 770 | 770 | $\underline{790}$ |
| 62-45 | 816 | **816** | 798 | 810 | 810 | **816** | **816** | **816** | **816** |
| 62-50 | 900 | 870 | 846 | $\underline{852}$ | $\underline{852}$ | 846 | 846 | $\underline{852}$ | $\underline{852}$ |
| 62-55 | 984 | 978 | $\underline{972}$ | $\underline{972}$ | $\underline{972}$ | $\underline{972}$ | $\underline{972}$ | $\underline{972}$ | $\underline{972}$ |
| 62-60 | 1062 | 1038 | $\underline{1044}$ | $\underline{1044}$ | $\underline{1044}$ | $\underline{1044}$ | $\underline{1044}$ | $\underline{1044}$ | $\underline{1044}$ |
| 62-65 | 1116 | 1104 | 1110 | **1116** | **1116** | 1110 | 1110 | **1116** | **1116** |
| 62-70 | 1188 | 1170 | 1122 | $\underline{1152}$ | $\underline{1152}$ | 1122 | 1122 | $\underline{1152}$ | $\underline{1152}$ |
| 62-75 | 1236 | 1200 | $\underline{1200}$ | $\underline{1200}$ | $\underline{1200}$ | $\underline{1200}$ | $\underline{1200}$ | $\underline{1200}$ | $\underline{1200}$ |
| 62-80 | 1284 | 1266 | $\underline{1254}$ | $\underline{1254}$ | $\underline{1254}$ | $\underline{1254}$ | $\underline{1254}$ | $\underline{1254}$ | $\underline{1254}$ |
| 64-40 | 575 | 570 | $\underline{570}$ | $\underline{570}$ | $\underline{570}$ | $\underline{570}$ | $\underline{570}$ | $\underline{570}$ | $\underline{570}$ |
| 64-45 | 650 | 645 | $\underline{645}$ | $\underline{645}$ | $\underline{645}$ | $\underline{645}$ | $\underline{645}$ | $\underline{645}$ | $\underline{645}$ |
| 64-50 | 730 | 715 | $\underline{715}$ | $\underline{715}$ | $\underline{715}$ | $\underline{715}$ | $\underline{715}$ | $\underline{715}$ | $\underline{715}$ |
| 64-55 | 825 | 805 | 805 | 805 | 805 | **825** | **825** | **825** | **825** |
| 64-60 | 915 | 910 | $\underline{910}$ | $\underline{910}$ | $\underline{910}$ | $\underline{910}$ | $\underline{910}$ | $\underline{910}$ | $\underline{910}$ |
| 64-125 | 1670 | 1635 | 1635 | 1635 | $\underline{1645}$ | 1635 | 1635 | 1635 | $\underline{1645}$ |
| 64-130 | 1680 | 1670 | $\underline{1660}$ | $\underline{1660}$ | $\underline{1660}$ | $\underline{1660}$ | $\underline{1660}$ | $\underline{1660}$ | $\underline{1660}$ |
| 99-30 | 173 | **173** | **173** | **173** | **173** | **173** | **173** | **173** | **173** |
| 99-35 | 241 | **241** | **241** | **241** | **241** | **241** | **241** | **241** | **241** |
| 99-40 | 299 | **299** | **299** | **299** | **299** | **299** | **299** | **299** | **299** |
| 99-45 | 367 | **367** | **367** | **367** | **367** | **367** | **367** | **367** | **367** |
| 101-50 | 181 | **181** | **181** | **181** | **181** | **181** | **181** | **181** | **181** |
| 101-60 | 243 | **243** | **243** | **243** | **243** | **243** | **243** | **243** | **243** |

Table 9: SET2 - H=7 and D=3

| S-T | OptVal | DVC | Init. | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|---|
| 30-65 | 240 | **240** | **240** | **240** | **240** | **240** | **240** | **240** | **240** |
| 30-70 | 260 | **260** | **260** | **260** | **260** | **260** | **260** | **260** | **260** |
| 30-73 | 265 | **265** | **265** | **265** | **265** | **265** | **265** | **265** | **265** |
| 30-75 | 270 | **270** | $\underline{260}$ | $\underline{260}$ | $\underline{260}$ | $\underline{260}$ | $\underline{260}$ | $\underline{260}$ | $\underline{260}$ |
| 30-80 | 280 | 275 | 275 | **280** | **280** | 275 | 275 | **280** | **280** |
| 30-85 | 285 | **285** | 280 | **285** | 280 | 280 | 280 | **285** | 280 |
| 30-65 | 610 | **610** | **610** | **610** | **610** | **610** | **610** | **610** | **610** |
| 30-75 | 670 | 650 | $\underline{650}$ | $\underline{650}$ | $\underline{650}$ | $\underline{650}$ | $\underline{650}$ | $\underline{650}$ | $\underline{650}$ |
| 30-80 | 710 | **710** | **710** | **710** | **710** | **710** | **710** | **710** | **710** |
| 30-85 | 740 | 710 | 690 | 690 | 700 | 690 | **740** | 690 | **740** |
| 30-90 | 770 | 740 | $\underline{730}$ | $\underline{730}$ | $\underline{730}$ | $\underline{730}$ | $\underline{730}$ | $\underline{730}$ | $\underline{730}$ |
| 30-95 | 790 | **790** | **790** | **790** | **790** | **790** | **790** | **790** | **790** |
| 30-100 | 800 | **800** | **800** | **800** | **800** | **800** | **800** | **800** | **800** |
| 30-105 | 800 | **800** | **800** | **800** | **800** | **800** | **800** | **800** | **800** |
| 62-45 | 816 | **816** | $\underline{804}$ | $\underline{804}$ | $\underline{804}$ | $\underline{804}$ | $\underline{804}$ | $\underline{804}$ | $\underline{804}$ |
| 62-50 | 900 | 870 | $\underline{870}$ | $\underline{870}$ | $\underline{870}$ | $\underline{870}$ | $\underline{870}$ | $\underline{870}$ | $\underline{870}$ |
| 62-55 | 984 | 954 | 948 | $\underline{954}$ | $\underline{954}$ | 948 | 948 | $\underline{954}$ | $\underline{954}$ |
| 62-60 | 1062 | 1056 | $\underline{1056}$ | $\underline{1056}$ | $\underline{1056}$ | $\underline{1056}$ | $\underline{1056}$ | $\underline{1056}$ | $\underline{1056}$ |
| 62-65 | 1116 | 1092 | 1080 | 1080 | $\underline{1092}$ | 1080 | 1080 | 1080 | $\underline{1098}$ |
| 62-70 | 1188 | 1170 | 1158 | $\underline{1170}$ | $\underline{1170}$ | 1158 | 1158 | $\underline{1170}$ | $\underline{1170}$ |
| 62-75 | 1236 | 1218 | 1206 | $\underline{1212}$ | $\underline{1212}$ | 1206 | 1206 | $\underline{1212}$ | $\underline{1212}$ |
| 62-80 | 1284 | 1260 | $\underline{1260}$ | $\underline{1260}$ | $\underline{1260}$ | $\underline{1260}$ | $\underline{1260}$ | $\underline{1260}$ | $\underline{1260}$ |
| 64-40 | 575 | 570 | $\underline{570}$ | $\underline{570}$ | $\underline{570}$ | $\underline{570}$ | $\underline{570}$ | $\underline{570}$ | $\underline{570}$ |
| 64-45 | 650 | 645 | $\underline{645}$ | $\underline{645}$ | $\underline{645}$ | $\underline{645}$ | $\underline{645}$ | $\underline{645}$ | $\underline{645}$ |
| 64-50 | 730 | 715 | 690 | $\underline{715}$ | $\underline{715}$ | 690 | 690 | $\underline{715}$ | $\underline{715}$ |
| 64-55 | 825 | **825** | 805 | **825** | **825** | 805 | 805 | **825** | **825** |
| 64-60 | 915 | 890 | 890 | $\underline{910}$ | $\underline{910}$ | 890 | 890 | $\underline{910}$ | $\underline{910}$ |
| 64-125 | 1670 | 1655 | $\underline{1655}$ | $\underline{1655}$ | $\underline{1655}$ | $\underline{1655}$ | $\underline{1655}$ | $\underline{1655}$ | $\underline{1655}$ |
| 64-130 | 1680 | 1675 | $\underline{1675}$ | $\underline{1675}$ | $\underline{1675}$ | $\underline{1675}$ | $\underline{1675}$ | $\underline{1675}$ | $\underline{1675}$ |
| 99-30 | 173 | **173** | **173** | **173** | **173** | **173** | **173** | **173** | **173** |
| 99-35 | 241 | **241** | **241** | **241** | **241** | **241** | **241** | **241** | **241** |
| 99-40 | 299 | **299** | **299** | **299** | **299** | **299** | **299** | **299** | **299** |
| 99-45 | 367 | **367** | **367** | **367** | **367** | **367** | **367** | **367** | **367** |
| 101-50 | 181 | **181** | **181** | **181** | **181** | **181** | **181** | **181** | **181** |
| 101-60 | 243 | **243** | **243** | **243** | **243** | **243** | **243** | **243** | **243** |

Table 10: SET2 - H=8 and D=4

| S-T | OptVal | DVC | Init. | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|---|
| 30-65 | 240 | **240** | **240** | **240** | **240** | **240** | **240** | **240** | **240** |
| 30-70 | 260 | **260** | **260** | **260** | **260** | **260** | **260** | **260** | **260** |
| 30-73 | 265 | **265** | **265** | **265** | **265** | **265** | **265** | **265** | **265** |
| 30-75 | 270 | **270** | **270** | **270** | **270** | **270** | **270** | **270** | **270** |
| 30-80 | 280 | **280** | 275 | 275 | 275 | 275 | 275 | 275 | 275 |
| 30-85 | 285 | **285** | **285** | **285** | **285** | **285** | **285** | **285** | **285** |
| 30-65 | 610 | **610** | **610** | **610** | **610** | **610** | **610** | **610** | **610** |
| 30-75 | 670 | 650 | 650 | 650 | 650 | 650 | 650 | 650 | 650 |
| 30-80 | 710 | **710** | 660 | 660 | 660 | 660 | 660 | 660 | 660 |
| 30-85 | 740 | **740** | 710 | **740** | 710 | 710 | 710 | **740** | 710 |
| 30-90 | 770 | 730 | 720 | 720 | 720 | 720 | 720 | 720 | 720 |
| 30-95 | 790 | 750 | 750 | **790** | **790** | 750 | 750 | **790** | **790** |
| 30-100 | 800 | 760 | 740 | 760 | 740 | 740 | 740 | 760 | 740 |
| 30-105 | 800 | **800** | 770 | 770 | 790 | 770 | 770 | 770 | 790 |
| 62-45 | 816 | 792 | 798 | 810 | 810 | **816** | **816** | **816** | **816** |
| 62-50 | 900 | 870 | 846 | 852 | 852 | 870 | 852 | 852 | 852 |
| 62-55 | 984 | 972 | 972 | 972 | 972 | 972 | 972 | 972 | 972 |
| 62-60 | 1062 | 1044 | 1044 | 1044 | 1044 | 1044 | 1044 | 1044 | 1044 |
| 62-65 | 1116 | **1116** | 1110 | **1116** | **1116** | 1110 | 1110 | **1116** | **1116** |
| 62-70 | 1188 | 1164 | 1152 | 1152 | 1152 | 1152 | 1152 | 1152 | 1152 |
| 62-75 | 1236 | 1200 | 1200 | 1200 | 1200 | 1200 | 1200 | 1200 | 1200 |
| 62-80 | 1284 | 1266 | 1266 | 1266 | 1272 | 1266 | 1266 | 1266 | 1272 |
| 64-40 | 575 | 570 | 570 | 570 | 570 | 570 | 570 | 570 | 570 |
| 64-45 | 650 | 645 | 645 | 645 | 645 | 645 | 645 | 645 | 645 |
| 64-50 | 730 | 715 | 715 | 715 | 715 | 715 | 715 | 715 | 715 |
| 64-55 | 825 | 805 | 805 | 805 | 805 | **825** | **825** | **825** | **825** |
| 64-60 | 915 | 910 | 910 | 910 | 910 | 910 | 910 | 910 | 910 |
| 64-125 | 1670 | 1645 | 1645 | 1645 | 1650 | 1645 | 1645 | 1645 | 1650 |
| 64-130 | 1680 | 1670 | 1675 | 1675 | 1675 | 1675 | 1675 | 1675 | 1675 |
| 99-30 | 173 | **173** | **173** | **173** | **173** | **173** | **173** | **173** | **173** |
| 99-35 | 241 | **241** | **241** | **241** | **241** | **241** | **241** | **241** | **241** |
| 99-40 | 299 | **299** | **299** | **299** | **299** | **299** | **299** | **299** | **299** |
| 99-45 | 367 | **367** | **367** | **367** | **367** | **367** | **367** | **367** | **367** |
| 101-50 | 181 | **181** | **181** | **181** | **181** | **181** | **181** | **181** | **181** |
| 101-60 | 243 | **243** | **243** | **243** | **243** | **243** | **243** | **243** | **243** |

Table 11: SET3: H=12 and D=4

| S-T | OptVal | DVC | Init. | 1-V | 2-V | 1-T | 2-T |
|---|---|---|---|---|---|---|---|
| 62-75 | 1236 | 1224 | 1200 | 1206 | $\underline{1218}$ | 1200 | 1200 |
| 62-80 | 1284 | 1278 | 1266 | 1266 | $\underline{1272}$ | 1266 | 1266 |
| 64-125 | 1670 | **1670** | 1655 | 1655 | $\underline{1660}$ | 1655 | 1655 |
| 64-130 | 1680 | 1675 | 1665 | 1665 | $\underline{1675}$ | 1665 | 1665 |
| 98-50 | 412 | 408 | $\underline{408}$ | $\underline{408}$ | $\underline{408}$ | $\underline{408}$ | $\underline{408}$ |
| 98-60 | 504 | **504** | 488 | **504** | **504** | 488 | 488 |
| 98-70 | 590 | 575 | $\underline{575}$ | $\underline{575}$ | $\underline{575}$ | $\underline{575}$ | $\underline{575}$ |
| 98-80 | 652 | 641 | 626 | **652** | 626 | 626 | 626 |
| 98-90 | 725 | 706 | 706 | **725** | 706 | **725** | $\underline{725}$ |
| 98-100 | 782 | 766 | 756 | $\underline{764}$ | 756 | 756 | 756 |
| 98-110 | 835 | **835** | 792 | $\underline{805}$ | 795 | 792 | 792 |
| 98-120 | 894 | 886 | 827 | $\underline{861}$ | 855 | 837 | 837 |
| 98-130 | 956 | 909 | 884 | $\underline{918}$ | 885 | 884 | 884 |
| 98-140 | 1013 | 954 | 930 | $\underline{948}$ | $\underline{948}$ | 930 | 930 |
| 98-150 | 1057 | 1042 | $\underline{1025}$ | $\underline{1025}$ | $\underline{1025}$ | $\underline{1025}$ | $\underline{1025}$ |
| 98-160 | 1114 | 1023 | 1018 | 1018 | $\underline{1022}$ | 1018 | 1018 |
| 98-170 | 1164 | 1077 | 1070 | 1070 | $\underline{1078}$ | 1070 | 1070 |
| 98-180 | 1201 | 1142 | 1126 | $\underline{1167}$ | $\underline{1167}$ | 1126 | 1126 |
| 98-190 | 1234 | 1176 | $\underline{1155}$ | $\underline{1155}$ | $\underline{1155}$ | $\underline{1155}$ | $\underline{1155}$ |
| 98-200 | 1261 | 1220 | 1193 | 1210 | $\underline{1213}$ | 1193 | 1193 |
| 98-210 | 1284 | 1235 | 1218 | 1220 | $\underline{1235}$ | 1218 | 1218 |
| 98-240 | 1306 | 1299 | 1292 | 1292 | $\underline{1303}$ | 1292 | 1292 |

Table 12: SET3 - H=14 and D=5

| S-T | OptVal | DVC | Init. | 1-V | 2-V | 1-T | 2-T |
|---|---|---|---|---|---|---|---|
| 62-75 | 1236 | 1224 | 1200 | 1206 | 1218 | 1200 | 1200 |
| 62-80 | 1284 | 1278 | 1266 | 1266 | 1272 | 1266 | 1266 |
| 64-125 | 1670 | **1670** | 1655 | 1655 | 1660 | 1655 | 1655 |
| 64-130 | 1680 | 1675 | 1665 | 1665 | 1675 | 1665 | 1665 |
| 98-50 | 412 | 408 | 408 | 408 | 408 | 408 | 408 |
| 98-60 | 504 | **504** | 488 | **504** | **504** | 488 | 488 |
| 98-70 | 590 | 575 | 575 | 575 | 575 | 575 | 575 |
| 98-80 | 652 | 641 | 626 | **652** | 626 | 626 | 626 |
| 98-90 | 725 | 706 | 706 | **725** | 706 | **725** | 725 |
| 98-100 | 782 | 766 | 756 | 764 | 756 | 756 | 756 |
| 98-110 | 835 | **835** | 792 | 805 | 795 | 792 | 792 |
| 98-120 | 894 | 886 | 827 | 861 | 855 | 837 | 837 |
| 98-130 | 956 | 909 | 884 | 918 | 885 | 884 | 884 |
| 98-140 | 1013 | 954 | 930 | 948 | 948 | 930 | 930 |
| 98-150 | 1057 | 1042 | 1025 | 1025 | 1025 | 1025 | 1025 |
| 98-160 | 1114 | 1023 | 1018 | 1018 | 1022 | 1018 | 1018 |
| 98-170 | 1164 | 1077 | 1070 | 1070 | 1078 | 1070 | 1070 |
| 98-180 | 1201 | 1142 | 1126 | 1167 | 1167 | 1126 | 1126 |
| 98-190 | 1234 | 1176 | 1155 | 1155 | 1155 | 1155 | 1155 |
| 98-200 | 1261 | 1220 | 1193 | 1210 | 1213 | 1193 | 1193 |
| 98-210 | 1284 | 1235 | 1218 | 1220 | 1235 | 1218 | 1218 |
| 98-240 | 1306 | 1299 | 1292 | 1292 | 1303 | 1292 | 1292 |

Table 13: SET4 - H=5 and D=2/3

| S - T - D | OptVal | LB | UB | DVC | Init. | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 98-20-2 | 247 | | | 247 | 240 | 240 | 240 | 240 | 240 | 240 | 240 |
| 98-30-2 | 385 | | | 385 | **385** | **385** | **385** | **385** | **385** | **385** | **385** |
| 100-35-2 | 157 | | | 151 | 151 | 151 | 151 | 151 | 151 | 151 | 151 |
| 100-40-2 | 210 | | | 210 | **210** | **210** | **210** | **210** | **210** | **210** | **210** |
| 100-45-2 | 266 | | | 266 | 254 | 254 | 254 | 254 | 254 | 254 | 254 |
| 98-20-3 | | 357 | 376 | 368 | 368 | 368 | 368 | 368 | 368 | 368 | 368 |
| 98-25-3 | | 495 | 568 | 524 | 524 | 524 | 524 | 524 | 524 | 524 | 524 |
| 100-30-3 | | 230 | 380 | 324 | 300 | 300 | 300 | 300 | 300 | 300 | 300 |
| 100-40-3 | | 299 | 493 | 383 | 366 | 383 | 383 | 387 | 387 | 383 | 383 |
| 100-45-3 | | 356 | 579 | 442 | 420 | 430 | 425 | 425 | 425 | 425 | 427 |

## B Rank of the Hotel Combinations of the Best Found Solutions

These tables show for each test instance and each solving method the rank of the hotel combination of the best found solution, which was determined after the initialization phase.

Table 14: SET1 - H=3 D=2

| S - T | TNFC | kMax | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|
| 30-65 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-70 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-73 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-75 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-80 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-85 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-65 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-75 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-80 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-85 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-90 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-95 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-100 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-105 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-45 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-50 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-55 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-60 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-65 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-70 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-75 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-80 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-40 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-45 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-50 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-55 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-60 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-125 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-130 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-30 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-35 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-40 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-45 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 101-50 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| 101-60 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 15: SET1 - H=4 and D=3

| S - T | TNFC | kMax | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|
| 30-65 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-70 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-73 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-75 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-80 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-85 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-65 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-75 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-80 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-85 | 16 | 16 | 5 | 3 | 1 | 3 | 5 | 3 |
| 30-90 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-95 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-100 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-105 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-45 | 12 | 12 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-50 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-55 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-60 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-65 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 8 |
| 62-70 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-75 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-80 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-40 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-45 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-50 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-55 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-60 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-125 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-130 | 16 | 16 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-30 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-35 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-40 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-45 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 101-50 | 12 | 12 | 1 | 1 | 1 | 1 | 1 | 1 |
| 101-60 | 12 | 12 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 16: SET1 - H=5 and D=4

| S - T | TNFC | kMax | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|
| 30-65 | 107 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-70 | 107 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-73 | 107 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-75 | 107 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-80 | 125 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-85 | 125 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-65 | 100 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-75 | 125 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-80 | 107 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-85 | 107 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-90 | 100 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-95 | 100 | 63 | 1 | 6 | 1 | 1 | 1 | 5 |
| 30-100 | 125 | 63 | 14 | 1 | 1 | 1 | 14 | 1 |
| 30-105 | 125 | 63 | 1 | 3 | 1 | 1 | 1 | 3 |
| 62-45 | 44 | 44 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-50 | 72 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-55 | 100 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-60 | 100 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-65 | 125 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-70 | 125 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-75 | 125 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-80 | 125 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-40 | 107 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-45 | 88 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-50 | 107 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-55 | 107 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-60 | 107 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-125 | 125 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-130 | 125 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-30 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-35 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-40 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-45 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 101-50 | 33 | 33 | 1 | 1 | 1 | 1 | 1 | 1 |
| 101-60 | 60 | 60 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 17: SET2 - H=7 and D=3

| S - T | TNFC | kMax | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|
| 30-65 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-70 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-73 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-75 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-80 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-85 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-65 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-75 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-80 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-85 | 49 | 49 | 1 | 12 | 1 | 13 | 1 | 12 |
| 30-90 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-95 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-100 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-105 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-45 | 42 | 42 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-50 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-55 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-60 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-65 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 26 |
| 62-70 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-75 | 49 | 49 | 4 | 15 | 1 | 1 | 4 | 15 |
| 62-80 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-40 | 45 | 45 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-45 | 47 | 47 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-50 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-55 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-60 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-125 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-130 | 49 | 49 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-30 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-35 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-40 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-45 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 101-50 | 14 | 14 | 1 | 1 | 1 | 1 | 1 | 1 |
| 101-60 | 17 | 17 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 18: SET2 - H=8 and D=4

| S - T | TNFC | kMax | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|
| 30-65 | 482 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-70 | 482 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-73 | 482 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-75 | 482 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-80 | 512 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-85 | 512 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-65 | 448 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-75 | 512 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-80 | 482 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-85 | 482 | 63 | 2 | 1 | 1 | 1 | 2 | 1 |
| 30-90 | 448 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 30-95 | 448 | 63 | 11 | 11 | 1 | 1 | 11 | 11 |
| 30-100 | 512 | 63 | 21 | 1 | 1 | 1 | 21 | 1 |
| 30-105 | 512 | 63 | 1 | 7 | 1 | 1 | 1 | 7 |
| 62-45 | 252 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-50 | 378 | 63 | 1 | 1 | 2 | 7 | 1 | 1 |
| 62-55 | 448 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-60 | 448 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-65 | 512 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-70 | 512 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-75 | 512 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 62-80 | 512 | 63 | 1 | 2 | 1 | 1 | 1 | 2 |
| 64-40 | 302 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-45 | 228 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-50 | 357 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-55 | 400 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-60 | 424 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 64-125 | 512 | 63 | 1 | 7 | 1 | 1 | 1 | 7 |
| 64-130 | 512 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-30 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-35 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-40 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 99-45 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 101-50 | 60 | 60 | 1 | 1 | 1 | 1 | 1 | 1 |
| 101-60 | 105 | 63 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 19: SET3 - H=12 and D=4

| S - T | TNFC | kMax | 1-V | 2-V | 1-T | 2-T |
|---|---|---|---|---|---|---|
| 62-75 | 1728 | 63 | 24 | 42 | 1 | 1 |
| 62-80 | 1728 | 63 | 1 | 11 | 1 | 1 |
| 64-125 | 1728 | 63 | 1 | 3 | 1 | 1 |
| 64-130 | 1728 | 63 | 1 | 13 | 1 | 1 |
| 98-50 | 39 | 39 | 1 | 1 | 1 | 1 |
| 98-60 | 161 | 63 | 1 | 1 | 1 | 1 |
| 98-70 | 276 | 63 | 1 | 1 | 1 | 1 |
| 98-80 | 892 | 63 | 7 | 1 | 1 | 7 |
| 98-90 | 1220 | 63 | 2 | 1 | 2 | 2 |
| 98-100 | 1296 | 63 | 11 | 1 | 1 | 1 |
| 98-110 | 1728 | 63 | 2 | 1 | 1 | 1 |
| 98-120 | 1728 | 63 | 36 | 1 | 12 | 1 |
| 98-130 | 1551 | 63 | 55 | 3 | 1 | 51 |
| 98-140 | 1551 | 63 | 1 | 1 | 1 | 1 |
| 98-150 | 1728 | 63 | 1 | 1 | 1 | 1 |
| 98-160 | 1728 | 63 | 1 | 11 | 1 | 1 |
| 98-170 | 1728 | 63 | 1 | 1 | 1 | 1 |
| 98-180 | 1728 | 63 | 1 | 1 | 1 | 1 |
| 98-190 | 1728 | 63 | 1 | 1 | 1 | 1 |
| 98-200 | 1728 | 63 | 25 | 1 | 1 | 1 |
| 98-210 | 1728 | 63 | 1 | 12 | 1 | 1 |
| 98-240 | 1728 | 63 | 1 | 1 | 1 | 1 |

Table 20: SET3 - H=14 and D=5

| S-T | TNFC | kMax | 1-V | 2-V | 1-T | 2-T |
|---|---|---|---|---|---|---|
| 62-75 | 32928 | 63 | 7 | 1 | 1 | 14 |
| 62-80 | 32928 | 63 | 1 | 11 | 1 | 1 |
| 64-125 | 38416 | 63 | 1 | 1 | 1 | 1 |
| 64-130 | 38416 | 63 | 1 | 1 | 1 | 1 |
| 98-50 | 26 | 26 | 1 | 1 | 1 | 1 |
| 98-60 | 442 | 63 | 1 | 1 | 1 | 1 |
| 98-70 | 1273 | 63 | 1 | 1 | 1 | 1 |
| 98-80 | 3053 | 63 | 1 | 1 | 1 | 1 |
| 98-90 | 5240 | 63 | 1 | 1 | 1 | 1 |
| 98-100 | 15272 | 63 | 2 | 1 | 1 | 1 |
| 98-110 | 10991 | 63 | 1 | 1 | 1 | 1 |
| 98-120 | 15423 | 63 | 1 | 1 | 1 | 1 |
| 98-130 | 24210 | 63 | 1 | 2 | 1 | 1 |
| 98-140 | 27440 | 63 | 1 | 31 | 1 | 1 |
| 98-150 | 38416 | 63 | 1 | 1 | 1 | 1 |
| 98-160 | 35672 | 63 | 1 | 1 | 1 | 1 |
| 98-170 | 38416 | 63 | 1 | 1 | 1 | 1 |
| 98-180 | 38416 | 63 | 1 | 6 | 1 | 1 |
| 98-190 | 38416 | 63 | 1 | 2 | 1 | 1 |
| 98-200 | 38416 | 63 | 1 | 1 | 1 | 1 |
| 98-210 | 38416 | 63 | 1 | 23 | 1 | 1 |
| 98-240 | 38416 | 63 | 1 | 21 | 1 | 1 |

Table 21: SET4 - H=5 and D=2/3

| S - T | TNFC | kMax | 1-V | 2-V | 1-T | 2-T | 1-VT | 2-VT |
|---|---|---|---|---|---|---|---|---|
| 98-20 | 4 | 4 | 1 | 1 | 1 | 1 | 1 | 1 |
| 98-25 | 5 | 5 | 1 | 1 | 1 | 1 | 1 | 1 |
| 100-35 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 100-40 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 100-45 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 98-20 | 19 | 19 | 1 | 1 | 1 | 1 | 1 | 1 |
| 98-25 | 25 | 25 | 1 | 1 | 1 | 1 | 1 | 1 |
| 100-30 | 7 | 7 | 1 | 1 | 1 | 1 | 1 | 1 |
| 100-40 | 7 | 7 | 1 | 1 | 1 | 1 | 1 | 1 |
| 100-45 | 7 | 7 | 2 | 1 | 1 | 1 | 1 | 1 |