



King's Research Portal

DOI: 10.1051/ro/2021148

Document Version Peer reviewed version

Link to publication record in King's Research Portal

Citation for published version (APA): M'Halla Ep Aounallah, R., & Ben Nejma, I. (2021). A beam search for the equality generalized symmetric traveling salesman problem. *RAIRO - Operations Research*, *55*(5), 3021-3039. https://doi.org/10.1051/ro/2021148

Citing this paper

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

General rights

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

•Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research. •You may not further distribute the material or use it for any profit-making activity or commercial gain •You may freely distribute the URL identifying the publication in the Research Portal

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

A beam search for the equality generalized symmetric traveling salesman problem

Ibtissem Ben Nejma

Logistics Department, Institut Superieur du Transport et de la Logistique de Sousse, University of Sousse, Sousse, Tunisia.

Rym M'Hallah

Engineering Department, Kings College London, Strand, London WC2R 2LS, UK.

Abstract

This paper studies the equality generalized symmetric traveling salesman problem (EGSTSP). A salesman has to visit a predefined set of countries. S/he must determine exactly one city (of a subset of cities) to visit in each country and the sequence of the countries such that s/he minimizes the overall travel cost. From an academic perspective, EGSTSP is very important. It is NP-hard. Its relaxed version TSP is itself NP-hard, and no exact technique solves large difficult instances. From a logistic perspective, EGSTSP has a broad range of applications that vary from sea, air, and train shipping to emergency relief to elections and polling to airlines' scheduling to urban transportation. During the COVID-19 pandemic, the roll-out of vaccines further emphasizes the importance of this problem. Pharmaceutical firms are challenged not only by a viable production schedule but also by a flawless distribution plan especially that some of these vaccines must be stored at extremely low temperatures. This paper proposes an approximate tree-based search technique for EGSTSP. It uses a beam search with low and high level hybridization. The low-level hybridization applies a swap based local search to each partial solution of a node of a tree whereas the high-level hybridization applies 2-Opt, 3-Opt or Lin-Kernighan to the incumbent. Empirical results provide computational evidence that the proposed approach solves large instances with 89 countries and 442 cities in few seconds while matching the best known cost of 8 out of 36 instances and being less than 1.78% away from the best known solution for 27 instances.

Email addresses: nejmalog@gmail.com (Ibtissem Ben Nejma),

Preprint submitted to RAIRO - Operations Research

September 20, 2021

rym.mhallah@kcl.ac.uk (Rym M'Hallah)

^{*}Corresponding author. Tel: +44-749-8063841.

Keywords: Generalized traveling salesman, beam search, symmetric traveling salesman, *k*-opt, Lin-Kernighan heuristic.

1. Introduction

Logistics in general and transportation in particular are the cornerstones of modern life. Their importance emanates from their multi-fold repercussions on the cost of goods, profit margins of transportation companies, clients' service quality, drivers' well being, and air pollution. In fact, they involve several parties: end clients, manufacturers, distributors, drivers, stock holders, etc. In addition, they require the scheduling of several interrelated tasks that are dynamic in nature and constrained in time and space. The economic and temporal constraints augment their complexity. Solving them requires the migration of tools from diverse disciplines including information technology, optimization, and vehicle routing.

Among the most widely studied transportation problems is the traveling salesman problem (TSP). A traveler has to visit a finite number of countries starting from one country and returning back to it, and visiting every country exactly once. The objective is to find a minimal cost route, where the cost can be total duration, travel distance, etc. TSP's importance emanates from its occurrence as a subproblem of complex real life problems in the transport of passengers/goods and in scheduling. For these problems, TSP identifies a minimal-cost itinerary for each salesman. For example, TSP is a special case of the equality generalized symmetric traveling salesman problem (EGSTSP), where the salesman chooses exactly one of many cities of a country to visit; i.e., a TSP with a covering constraint.

Formally, consider a set of nodes that are divided into clusters. EGSTSP searches for the shortest route that visits exactly one node from every cluster starting and ending at the same cluster. EGSTSP is more difficult than TSP because of the combinatorial aspect added by the sizes of the clusters. EGSTSP occurs in several real-life applications such as maritime ship routing, distribution of medical supplies, urban waste management, telecommunication networks, logistics, rapid post dispatching, VLSI, circuit designs, and in laser cutting to determine the trajectory of a laster cutter [18, 21]. During the COVID-19 pandemic, EGSTSP has drawn a lot of attention. With reduced air-traffic and disrupted logistic chains, the procurement and dispatching of goods to confined customers and isolated cities has become a true challenge. In addition, the availability of a vaccine raises the issue of its fair distribution and of health care equity. Some of these vaccines impose

a cold chain that can't be broken. In such cases, optimizing the distribution plan is of prime importance. This optimization is equivalent to solving a large scale EGSTSP. Evidently, its exact solution may be challenging. However, the continuous advancement of the computing technologies provides near-optimal solutions to such difficult problems. They are allowing approximate methods to undertake a more extensive search; thus obtaining nearer-global optima in shorter times.

EGSTSP has been solved by exact approaches (such as dynamic programming, branch and bound, and branch and cut), and by approximate ones such as local improvement heuristics (*k*-opt, swap, insertion, etc.), and meta-heuristics (tabu search, ant colony, genetic algorithms, etc.). This paper proposes a new approximate hybrid approach for the EGSTSP. Hybrid heuristics have identified the best known solutions to several complex combinatorial optimization problems. They are powerful search methods because they tackle two competing goals: exploration and exploitation. Exploration is a diversification of the search. It investigates the solution space in order to determine the part that has a higher chance of containing the global optimum. Exploitation refines (or intensifies) the search on the part of the space that has a high potential of containing the global optimum.

The proposed hybrid heuristic is a beam search (BS) (i.e., a truncated branch and bound) that is augmented with improvement techniques. It ensures exploration via a standard width-first BS and exploitation via local search heuristics. BS strives for global optimization while local search heuristics strive for local optimization in the global optimum's neighborhood. That is, BS can be assimilated to evolution while local search to learning. Generally, synchronization of evolution and learning yields efficient hybrid heuristics. Specifically, the proposed hybrid BS embeds

- a low-level hybridization, which addresses the functional composition of BS by subjecting the partial solution at each node to a local search; and
- a high level hybridization that maintains BS self containing by subjecting the incumbent of BS to a k-opt type of search.

To the best of the authors' knowledge, this is the first application of BS to EGSTSP. In addition, the hybridization explores the success of local search to assess the nodes of the tree and to estimate their potential. It subsequently chooses the nodes with the best potential to branch on and prunes the non-promising ones; thus, it explores the search space's parts that contain near-global optima while it discards the others. It then applies a 2-opt, a 3-opt or the notorious improved Lin-Kernighan (LK) heuristic [9] to its incumbent. The computational investigation provides computational evidence of the good performance of the hybridized BS within a reduced runtime. Its deviation from the best known solution is less than 0.0578% for half of the instances and less than 1.78% for three quarters of them.

Section 2 defines the problem. Section 3 reviews the literature on EGSTSP. Section 4 details the proposed approaches. It presents the algorithm of a standard BS, its adaptation to EGSTSP, the low-level hybridized BS, and the high-level hybridized BS when applying each of the three local improvement methods: 2-Opt, 3-Opt and LK. Section 5 presents the computational results, which assess the efficiency of the methods in terms of solution quality and runtime and highlights the utility of the hybridizations. Finally, Section 6 summarizes the findings and gives potential research extensions.

2. Problem Definition

EGSTSP is an NP-hard combinatorial optimization problem. It consists of finding the optimal path of a salesman who has to travel through a set of countries while visiting exactly one city from each country and visiting every country once. The optimal path minimizes the total traveled cost. Hence, the salesman must determine for each country the city s/he will visit and the order of visit of the countries. EGSTSP is more complex than TSP. For TSP, each country consists of a single town while EGSTSP has the additional complexity of choosing a city from each country. Because it extends TSP, which is NP-hard, EGSTSP is also NP-hard.

Herein, we define EGSTSP using the notation of [6, 7] and present their integer linear program (ILP). Formally, consider a complete non-oriented graph G = (N, E) where $N = \{1, \ldots, n\}$ is a set of nodes that are divided into m mutually exclusive clusters C_h , $h = 1, \ldots, m$, and $m \ge 3$. E = $\{[i, j] : i \in N, j \in N, i \ne j\}$ denotes the set of edges e connecting pairs (i, j) of distinct nodes $i \in N$ and $j \in N$, $i \ne j$. The cost of traveling through edge $e \in E$ is d_e . This cost may be assimilated to a linear function of the Euclidean distance between i and j. The objective of EGSTSP is to determine a minimal cost cycle $T \subseteq E$ such that T includes exactly one city from each cluster, and each cluster is visited once.

To define the ILP model of EGSTSP, we introduce the following notation. For a subset $S \subseteq N$, $E(S) := \{[i, j] \in E : i \in S, j \in S\}$ denotes the set of edges with both endnodes in S and $\delta(S) := \{[i, j] \in E : i \in S, j \notin S\}$ the set of edges with exactly one end node in S. For simplicity, we denote $\delta(\{v\})$ by $\delta(v)$, for $v \in N$.

ILP uses two types of binary variables: $x_e = 1$ if the salesman travels through edge $e \in E$ and 0 otherwise, and $y_v = 1$ if the salesman visits node $v \in N$ and 0 otherwise. Using the aforementioned notation and these two sets of binary variables, EGSTSP can be formulated as follows.

$$\min z = \sum_{e \in E} d_e x_e \tag{1}$$

s.t.
$$\sum_{e \in \delta(v)} x_e \le 2y_v \qquad \qquad v \in N$$
(2)

$$\sum_{v \in C_k} y_v = 1 \qquad \qquad k = 1, \dots, m \quad (3)$$

$$\sum_{e \in \delta(S)} x_e \ge 2(y_i + y_j - 1) \quad S \subset N, 2 \le |S| \le n - 2, i \in S, j \in N \setminus S$$
(4)

$$x_e \in \{0,1\} \qquad e \in E \tag{5}$$

$$y_v \in \{0,1\} \qquad \qquad v \in N \tag{6}$$

The objective function, given by Equation (1), minimises the total travel cost. Equation (2) preserves the flow through every node. A node is visited if it has both a predecessor and a successor node; therefore the righthand side is 2; otherwise, the righthand side must be zero. Equation (3) ensures that the tour includes exactly one city from each cluster. Equation (4) guarantees the connectivity of the solution: Each cut separating two visited nodes i and j must be crossed at least twice. Finally, Equations (5) and (6) determine the nature of the decision variables.

Because EGSTSP is NP-hard, solving large instances of EGSTSP using ILP is difficult. Herein, we are interested in efficiently solving large instances of EGSTSP using heuristic methods, and in comparing the heuristics' solutions to the ILP results that are readily available in the literature (and given by z^{lit} in the computational section and in the Appendix).

3. Literature Review

Small instances of the equality generalized TSP (EGTSP) were solved exactly using dynamic programming [26], branch and bound [14], and branch and cut [6]. Large instances have been tackled approximately; for example, Noon and Bean [20] applied the TSP's closest neighborhood heuristic. Lien et al. [16] assimilated EGTSP to a TSP whose number of nodes is three times as large as the number of clusters. Dimitrijevic and Saric [5] developed an alternative transformation that had fewer nodes; i.e., using twice as many nodes as the number of clusters of the original EGTSP. Ben-Arieh et al. [2] opted for a transformation that had as many nodes as the number of clusters of EGTSP using the 'exact' Noon–Bean, and two modifications of the non-exact Fischetti–Salazar–Toth transformation. Helsgaun [9] transformed EGSTSP into a classic TSP and applied LK to the transformed TSP. Karapetyan and Gutin [11] proposed an LK heuristic for EGSTSP. Smith and Imeson [25] applied an iterative remove and insert heuristic for EGSTSP. They opted for three insertion mechanisms: the furthest node, the cheapest, and random insertion. Karapetyan and Gutin [12] also designed a large neighborhood search for EGSTSP. Renaud et al. [22] proposed an Initialization, Insertion and Improvement heuristic that Renaud and Boctor [23] further generalized. Khachai and Neznakhina [13] developed a dynamic programming based heuristic for EGSTSP.

Another surge of the EGSTSP literature came from hybrid approaches. Ardalan et al. [1] hybridized the Imperialist Competitive Algorithm with a local search. Lawrence and Daskin [15] hybridized a random key genetic algorithm with a local search. Their algorithm is quite fast. It identifies its best solution within the first two or three iterations. Its good performance is due to the utility of the local search in identifying the best solution. However, their algorithm is outperformed by the mimetic algorithm of Gutin et al. [8], who combined the advantages of genetic algorithms and local search. Chira et al. [4] designed a "sensible" and colony system that makes the ants sensitive to the pheromone level in their trail; thus, explore the most promising regions of the search space. Yang et al. [28] augmented ant colony optimization to EGSTSP with a mutation mechanism and a local search. They showed the importance of the local search, in particular, for instances with fewer than 200 nodes. Bontoux et al. [3] proposed a mimetic algorithm whose crossover operator is based on a large neighborhood search.

Different variants of EGSTSP have appeared recently. Sundar and Rathinam [27] applied a branch and cut and Zhou and Brian [30] extended Christofide's TSP algorithm to the multi-depot EGSTSP where there are several travelers; each departing from a different depot (node). Mestria [19] considered the clustered traveling salesman problem, where all nodes of a cluster must be visited in a contiguous manner. The author hybridized a variable neighborhood random descent with local search (for intensification) and with a greedy randomized adaptive search (for diversification). This latter consists of a constructive heuristic and a perturbation method. The author applied several variable neighborhood structures, in a random order. Jian et al. [10] proposed a hybrid genetic ant colony algorithm for the multiple TSP, where each salesman departs from a specific depot and returns to it. Yuan et al. [29] studied the generalized TSP with time windows, where arrival to a city must occur within a time window. They proposed two integer linear programs and valid inequalities that are separated dynamically within a branch-and-cut algorithm. They initiated their branch and bound from a feasible solution built via a simple heuristic. They solved instances with up to 30 clusters within a one-hour runtime. Salman et al. [24] imposed precedence constraints on EGSTSP, developed a new branching rule, and adapted some existing bounds to the problem.

This literature review suggests that EGSTSP was never tackled via BS. It further suggests that hybridization is a key factor in the success of most approaches to TSP related problems. To explore these findings, this paper proposes a hybrid BS that employs local search at each node and applies a k-opt type of search to the incumbent.

4. Proposed Approaches

We efficiently solve EGSTSP using hybridized BS-based algorithms. BS is a truncated tree search. It avoids exhaustive enumeration by branching on a subset of elite nodes, believed to lead to the optimum. They usually have minimal fitness values, which are either the cost of their partial solutions or their upper bounds. At each iteration, ω nodes are selected for branching, where ω is the beam width. The other nodes are permanently discarded, and no backtracking is performed. We enhance the performance of BS by hybridizing it at two levels. The low-level hybridization adds a local search phase at each node of the BS tree. The high-level hybridization applies 2opt, 3-opt or LK heuristics to the best solution that BS obtains. Section 4.1 describes a standard BS. Section 4.2 explains our adaptation of BS to EGSTSP. Sections 4.3 and 4.4 present the low- and high-level hybridization.

4.1. Standard Beam Search

The pseudo code of a standard BS is given in Figure 1. It consists of an initialization step, an iterative step and a stopping criterion. The initialization step declares the set \mathcal{N} of current nodes of the tree to the root node μ_0 and the set \mathcal{M} of offspring nodes to the empty set. When an initial feasible solution \mathbf{x} is available, this step further sets the incumbent \mathbf{x}^* and its value $z^* = z(\mathbf{x}^*)$ to, respectively, this initial solution \mathbf{x} and its objective function value. When an initial feasible solution is not available, the upper bound z^* is set to ∞ .

The iterative step chooses a node from \mathcal{N} , and sets it as the current node. It branches out of the current node, and adds all new nodes to \mathcal{M} except for leaves. Leaves constitute feasible solutions; thus, are candidate solutions. A leaf becomes the incumbent whenever its cost is less than z^* .

Initialization

Set $\mathcal{N} = \{\mu_0\}, \ \mathcal{M} = \emptyset.$

If an initial feasible solution \mathbf{x} whose cost $z(\mathbf{x})$ is available, set the incumbent $\mathbf{x}^* = \mathbf{x}$, and its objective function value $z^* = z(\mathbf{x})$; otherwise, set $z^* = \inf$.

Iterative step

- 1. Choose a node μ of \mathcal{N} ; branch out μ ; and insert the created nodes (i.e., the offsprings of μ) into \mathcal{M} .
- 2. If a node μ of \mathcal{M} is a leaf, then
 - compute its objective function value z_{μ} ;
 - if $z_{\mu} < z^*$, update z^* and the incumbent solution;
 - remove μ from \mathcal{M} .
- 3. Assess the potential of each node of \mathcal{M} .
- 4. Rank the nodes of \mathcal{M} in a non-descending order of their values.
- 5. Insert the min{ δ , $|\mathcal{M}|$ } best nodes of \mathcal{M} into \mathcal{N} ; and set $\mathcal{M} = \emptyset$.

Stopping condition

If $\mathcal{N} = \emptyset$, stop; otherwise, go to the *iterative step*.

Figure 1: A standard BS.

The iterative step appends the ω smallest-cost nodes of \mathcal{M} to \mathcal{N} and reinitializes \mathcal{M} to the empty set. This process is reiterated until no further branching is possible; that is, till $\mathcal{N} = \emptyset$. When applying a width-first BS, the nodes of \mathcal{N} belong to the same level of the tree.

4.2. Proposed Beam Search

This section presents our proposed BS-based method BS₀ for EGSTSP. BS₀ identifies a least cost ordering of the clusters. It assimilates the nodes of the tree to partial solutions (i.e., ordered subsets of C), and branching out of a node to augmenting it with an additional cluster. Its tree starts at the root node (i.e., level $\ell = 0$) with an empty tour, and has at most m levels. A partial solution s^{ℓ} corresponding to a node at level $\ell, \ell =$ $1, \ldots, m$, is a sequence of cities $i^1, i^2, \cdots, i^{\ell}$ all belonging to N and to different clusters. As all tree-search techniques, BS₀ has three major steps: branching, assessment, and selection. The branching of a node of the tree corresponds to appending a cluster to the partial solution of that node. That is, out of a node of level ℓ emanate $m-\ell$ branches; each leading to a different cluster. A node inherits the path of its parent, and appends a cluster to the end of its parent's path. Specifically, branching out of the node corresponding to s^{ℓ} consists in appending a city from a non-visited cluster to s^{ℓ} .

The assessment of the cost of a newly created node s^{ℓ} is based on a straightforward / simple lower bound and on an upper bound. The lower bound is the cost $z_{s^{\ell}}$ of the partial solution s^{ℓ} . It is the sum of the travel costs between the successive nodes of s^{ℓ} :

$$z_{s^{\ell}} = d_{i^1, i^2} + d_{i^2, i^3} + \ldots + d_{i^{\ell-2}, i^{\ell-1}} + d_{i^{\ell-1}, i^{\ell}}.$$

It is the sum of its parent node's cost $z_{s^{\ell-1}} = d_{i^1,i^2} + d_{i^2,i^3} + \ldots + d_{i^{\ell-2},i^{\ell-1}}$ and of the travel cost $d_{i^{\ell-1},i^{\ell}}$ from its parent node to the appended cluster. The upper bound is a total-cost of a complete solution constructed by iteratively appending the closest city of a 'not yet assigned' cluster to the partial solution s^{ℓ} .

At a given level ℓ of the tree, the selection chooses the ω best nodes among all generated child nodes for further branching at the next level $\ell + 1$ of the tree. These iterative branching, evaluation and selection processes are repeated until $\ell = m$; that is, until all clusters are visited. Herein, BS₀ is started with a feasible solution obtained via a greedy heuristic that chooses arbitrarily the first city i^1 and iteratively appends the closest city from a non-visited cluster.

In summary, BS_0 is a constructive approach that starts at the root node with an empty tour and appends a cluster at each level of the tree. It stops when the tour has m clusters visited. It has an $O(\omega m)$ worst case time complexity. Thus, our transformation of EGSTSP into TSP is less complex than competing transformations. It maintains m < n nodes whereas TSP considers n nodes.

4.3. Enhanced Beam Search

The low-level hybridized BS, denoted hereafter as BS₁, subjects each partial solution s^{ℓ} obtained at a node of a level ℓ , $\ell = 3, \ldots, m$, of the tree to a local search. The local search is simple but efficient. It preserves the order of the clusters in s^{ℓ} but changes the selected node of one or more clusters. It chooses the 'best' city among all nodes of every cluster of the partial solution s^{ℓ} . At a level $\ell \in \{3, \ldots, m\}$, BS generates $m - \ell$ nodes. Let s^{ℓ} be one of these nodes and let $s^{\ell} = ([1], \ldots, [\ell])$, where [i] denotes the *i*th cluster of the tour. The local search iterates for $h = [2], \ldots, [\ell - 1]$. It fixes the partial paths $[1], \ldots, [h-1]$ and $[h+1], \ldots, [\ell]$, and iterates through all the cities v of cluster C_h . It retains the city $v^* \in C_h$ that minimizes the distance from [h-1] to v to [h+1]; i.e.,

$$d_{[h-1]v^*} + d_{v^*[h+1]} = \min_{v \in C_h} \{ d_{[h-1]v} + d_{v[h+1]} \}.$$

When applied to a node s^{ℓ} , the local search has $O(\ell c)$ complexity $(c = \bar{c})$, where $\bar{c} = \max_{h=1,\dots,m} \{|C_h|\}$ is the maximum number of cities among all clusters.

Because it is applied to all $\sum_{\ell=3}^{m} \ell(m-\ell)$ nodes of the tree, the local search increases the complexity of BS₁ to at worst $O(\omega m^2 \bar{c})$. Yet, it allows BS₁ to attenuate the myopic nature of BS₀; i.e., BS₀ may miss the global optimum when it selects the ω best nodes of a level and permanently prunes the others.

4.4. High-Level Hybridized Beam Search

The high level hybridized BS, denoted BS., $\cdot = 2, 3, 4$, applies a 2-Opt, a 3-Opt, or LK heuristic to the best solution obtained by BS₁. Because the hybridization is high-level, the worst time complexity of BS., $\cdot = 2, 3, 4$ is the sum of the complexity of BS₁ and of the adopted hybridization approach.

The 2-Opt has an $O(m^2)$ complexity where m is the number of clusters of the tour. It chooses two clusters of the tour randomly and reverses the flow between them. It is repeated as long as the solution is improved. For instance, consider a tour $[1], [2], \ldots, [i-1], [i], [i+1], \ldots, [j-1], [j], [j+1], \ldots, [m], [1]$, where [i] denotes the $[i]^{th}$ cluster of the tour. When 2-Opt chooses randomly clusters [i] and [j], it generates the new solution $[1], [2], \ldots, [i-1], [j], [j-1], \ldots, [i+1], [i], [j+1], \ldots, [m], [1].$

The 3-Opt has an $O(m^3)$ complexity where m is the number of clusters of the tour. For a tour $[1], [2], \ldots, [i-1], [i], [i+1], \ldots, [j-1], [j], [j+1], \ldots, [\kappa-1], [\kappa], [\kappa+1], \ldots, [m], [1], 3$ -Opt chooses randomly three clusters [i], [j] and $[\kappa]$ of the tour, and generates the new solution $[1], [2], \ldots, [i], [\kappa], [\kappa-1], [j+1], [i+1], \ldots, [j], [\kappa+1], \ldots, [m], [1]$. It repeats this process as long as the solution is improved.

LK yields near-global optima when started from a large number of initial solutions. Any perturbation of its best solution causes increases of the order of 10 to 15% of its best cost. It is one of the best heuristics for the symmetric TSP because of its adaptive nature. Indeed, it swaps a number of partial sequences of the tour. This number is not predetermined; yet, it offers a good tradeoff between solution quality and runtime. While 2-opt and 3-opt break 2 and 3 edges of the tour, LK chooses the number of edges to be broken such that this number yields a minimal cost tour. In this sense, LK may be perceived as a variable-k exchange of k edges. It chooses k links to exchange and tests the utility of exchanging k + 1 links. (Initially k = 2.) Any exchange must generate a feasible neighbor. Its utility is assessed via the difference of the costs of the current solution and its neighbor. It is only adopted when it reduces the current solution's cost. LK marks the exchanged edges yielding the best net cost reduction as permanent and prohibits their elimination for a number of iterations by inserting them into a tabu list. When the exploration of exchanging k + 1 links reduces the incumbent's cost, LK updates the incumbent, and reduces k; otherwise, it increases k. LK stops when the incumbent can no longer be improved. Even though the complexity of LK is not well determined in the literature, our implementation has a worst time complexity of $O(m^5)$: It binds k to 5.

5. Computational Results

The computational investigation assesses the performance of hybridization in general, and of its type, in particular, on the solution quality and on the runtime of BS. For this purpose, it uses five versions of BS:

 \mathbf{BS}_0 A standard width-first beam search of beam width ω ,

 \mathbf{BS}_1 \mathbf{BS}_0 augmented with a local search at each node of the tree,

 \mathbf{BS}_2 BS₁ with its best solution subject to a 2-opt,

 \mathbf{BS}_3 BS₁ with its best solution subject to a 3-opt, and

 \mathbf{BS}_4 BS₁ with its best solution subject to the LK heuristic with k up to 5.

It applies these five versions (coded in C and run on an Intel Core i3-4030U, 1.90 GHz, 4GB RAM) to 36 benchmark instances of EGSTSP, all available at http://www.cs.rhul.ac.uk/home/zvero/GTSPLIB/. Let z^{lit} be the best known solution, and $z_{\omega}^{BS.}$, $\cdot = 0, 1, 2, 3, 4$ the corresponding BS. solution value, for a beam width $\omega = 1, 2, 3, 4, 5, 10$, obtained within runtime t_{ω} (expressed in seconds). For this solution, the percent optimality gap $\Delta_{\omega}^{\cdot} = 100\% \frac{z_{\omega} - z^{lit}}{z^{lit}}$. Herein, we analyze the results, reported in Appendix A, focusing on the utility of the low- and high-level hybridization of BS. We then conclude with some useful remarks.



Figure 2: Mean runtime of BS_0, \ldots, BS_4 as a function of beam width ω

5.1. Utility of the Low Level Hybridization

First, we compare the runtime and solution quality of BS_0 to that of BS_1 ; that is, of BS without and with local search at each node. (cf. Tables A.1 and A.2 for the detailed results.) We undertake this comparison to highlight the importance of the low-level hybridization undertaken at each node of each level ℓ of the search tree.

Figure 2, which displays the mean runtime of BS₀,..., BS₄, suggests that the mean runtime of BS increases linearly as a function of the beam width ω . Its average runtime (in seconds) can be estimated as a linear function of ω : $\bar{t}^0 = 0.5454\omega - 0.0459$ and $\bar{t}^1 = 0.5473\omega + 0.0080$, with 99.03% and 99.98% respective coefficients of determination. This behavior is expected as a larger beam width requires more evaluations of partial solutions, of bounds, of sorting, stocking, and retrieving.

Figure 3, which displays box plots of the observed run times, further clarifies this tendency. Yet, it stipulates that the local search does not increase the run time. A statistical paired t-test infers that there is no difference between the mean run times of BS₀ and BS₁ at any level of significance while a paired statistical test infers that the mean Δ_{BS_1} is less than the mean Δ_{BS_0} at any level of significance and that the mean difference $\Delta_{BS_0} - \Delta_{BS_1}$ has a 4.84% point estimate a 4.19% lower bound of a 95% confidence interval. This difference is due to the local search, which enhances the search of BS,



Figure 3: Box plots of observed run times of $BS_0, \ldots BS_4$ as a function of beam width ω

by investigating the neighborhood of the partial solution at each node. In fact, $\Delta_{BS_0} > \Delta_{BS_1}$ for all tested instances and for all beam widths. In addition, the average percent deviation $100\% \frac{z_{\omega}^1 - z_{\omega}^0}{z_{\omega}^0}$ is of the order of 26%; further highlighting the importance of the local search undertaken by BS₁ at every node. Because BS₁ is superior to BS₀ in terms of solution quality while being equally good in terms of runtime, it can be inferred that BS₁ is better than BS₀.

Figure 4 displays the box plots and means of the percent deviation of the solutions of BS., $\cdot = 0, \ldots, 4$, from z^{lit} . Zooming on the box plots and means of BS₀ and BS₁, we detect a seemingly counter-intuitive behavior for small ω . Increasing ω from 1 to 4 does not decrease Δ_{BS_0} and Δ_{BS_1} . This is most likely because it makes BS choose, at a level ℓ , partial solutions that –despite their good quality at level ℓ – do not lead to near-optima. That is, the diversification brought up by the larger beam width focuses on areas of the search space that do not contain the global optimum. The local search undertaken at each node does not mitigate this glitch. On the other hand, increasing ω beyond 5 overcomes this issue. Setting $\omega = 10$ allows BS to obtain solutions that are closer to the global optimum. That is, it makes BS investigate areas of the search space that contain near-global optima. This highlights the importance of the choice of the partial solutions at a level ℓ in



Figure 4: Box plots of percent deviation of the solutions of BS_0, \ldots, BS_4 as a function of beam width ω

order to direct the search toward the most promising regions. In this sense, the local search provides a lookahead strategy that helps BS judiciously choose its partial solutions.

5.2. Utility of the High Level Hybridization

Second, we compare the performance of BS_2 , BS_3 , and BS_4 . This comparison highlights the important impact of the high level hybridization, which requires a negligible additional runtime. (cf. Tables A.3 - A.5 for the detailed results.)

As Figure 4 reveals, the improvements of the solution quality due the high-level hybridization are much larger than their counterparts due to the low-level hybridization, regardless of the beam width. These improvements occur at no additional runtime cost except for the last three instances when run with BS₄ and a beam width $\omega = 10$. These instances are marked as outliers in Figure 3, which displays the box plots and means of the observed run times of BS₀ - BS₄. For all beam widths, the mean run time of any of the approaches is larger than its median; signaling the existence of outlier cases, corresponding to the last three instances. Despite the presence of these outliers, which drive the run time of BS₄ up for $\omega = 10$, paired t-tests infer that there is no statistical evidence to claim that the mean run time of



Figure 5: Mean percent deviations of the solutions of BS_0, \cdots, BS_4 from z^{lit} as a function of beam width ω

any pair of hybridized versions of BS are different at a 5% significance level.

The lack of exploitation and of exploration of the search space makes BS_0 obtain better results for larger beam widthes. This behavior persists for BS_1 , which benefits from a local search at each of its nodes, and for BS_2 , which benefits from an intensified 2-opt search around its best solution. However, for BS_3 and BS_4 , the 3-opt and the LK intensification makes BS obtain its best solutions using a beam width $\omega = 3$, with a mean runtime less than 2 seconds. This is confirmed by Figures 2 and 5, which display respectively the mean percent deviation from z^{lit} and mean runtime as a function of beam width for BS_0 to BS_4 .

5.3. Remarks

LK is known to obtain good results when initialized from several random initial solutions. The proposed approach BS₄ provides evidence that it is possible to generate initial solutions for LK in a more systematic manner. Furthermore, the results infer that BS₃ with a beam width $\omega = 3$ yields, on average, better results than the other considered beam searches. However, it remains true that the incumbent of BS₁ can be subjected to three types of searches 2-opt, 3-opt, and LK, at a negligible additional runtime. In fact, there is no statistical difference between the runtime of BS₁ and BS., $\cdot = 2, 3, 4$; implying that the bulk of their runtime is caused



Figure 6: Percent deviation of BS solutions from best known ones

by the BS component. Finally, even though $\omega = 3$ yields in general the best performance, running BS₁ with different beam widths constitutes a good diversification strategy. Using these two additional aforementioned intensification and diversification mechanisms reduces the percent deviation gaps of the BS solution to those observed in the literature; matching the best solution in 22.22% of the instances, and averaging a 0.01344% deviation. The mean should be interpreted with care as it is affected by two outlier values, recorded for instances 40kroA200 and 80rd400, as shown in Figure 6. These outliers are clearly depicted in Figure 7, which displays the resulting box plot of percent deviations for this BS. The corresponding five-point summary of the percent deviation is (Minimum=0, Q1=0.00063, Q2=0.00578, Q3=0.01779, Maximum=0.07027), where Q1, Q2 and Q3 are the first, second and third quartiles. Ignoring the two outlier instances brings the largest deviation over the other 34 instances to 0.03925% and its average to 0.01020%.

6. Conclusion

This paper addressed EGSTSP via a beam search that obtains good solutions for large beam widths. However, to avoid the exponential increase



Figure 7: Box plot of percent deviation of BS solutions from best known ones

of runtime associated with branch and bound, we opted for both a lowand a high-level hybridization of the beam search. First, we performed a local search at each node of the tree. This local search acts as a lookahead strategy. It allows the beam search to retain the partial solutions that could lead to near-global optima in lieu of selecting the lowest cost partial solutions. This local search improved the performance of the beam search without affecting its runtime. Second, we subjected the best solution of the beam search to each of three local search operators: 2-Opt, 3-Opt and Lin-Kernighan. This high level hybridization further improved the solution quality of the standard beam search by up to 70% without affecting its runtime. Applying the three search operators to the incumbent offers BS more exploration and exploitation power. The proposed hybridization can be applied to different variants of traveling related problems including vehicle routing, dial-a-ride, and delivery with time windows. Other types of search techniques can also be considered such as simulated annealing, variable neighborhood search, adaptive, and data-driven techniques.

References

 Ardalan A., Karimi S., Poursabzi O., Naderi B., 2015, A novel imperialist competitive algorithm for generalized traveling salesman problems. Applied Soft Computing 26:546-555.

- [2] Ben-Arieh D., Gutin G., Penn M., Yeo A., Zverovitch A., 2003, Transformations of generalized ATSP into ATSP. Operations Research Letters 31:357-365.
- [3] Bontoux B., Artigues C., Feillet D., 2010, A memetic algorithm with a large neighborhood crossover operator for the generalized traveling salesman problem. Computers & Operations Research 37:1844-1852.
- [4] Chira C., Pintea C.M., Dumitrescu D., 2007, Sensitive ant systems in combinatorial optimization. Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques, KEPT, Cluj-Napoca (Romania), 185–192.
- [5] Dimitrijevic V., Saric Z., 1997, An efficient transformation of the generalized traveling salesman problem into the traveling salesman problem. Digraphs, Information Sciences 102: 105–110.
- [6] Fischetti M., Salazar-González J.J., Toth P., 1997, A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. Operations Research 45:378–394.
- [7] Fischetti M., Salazar-González JJ., Toth P. (2007) The generalized traveling salesman and orienteering problems. In: Gutin G., Punnen A.P. (eds) The traveling salesman problem and its variations. Combinatorial Optimization, vol 12. Springer, Boston, MA.
- [8] Gutin G., Karapetyan D., Krasnogor N., 2008, Memetic algorithm for the generalized asymmetric traveling salesman problem. Studies in Computational Intelligence 129:199–210.
- [9] Helsgaun K., 2015, Solving the equality generalized traveling salesman problem using the Lin-Kernighan-Helsgaun-algorithm. Mathematical Programming Computation, 17(3):1–19.
- [10] Jiang C., Wan Z., Peng Z., 2020, A new efficient hybrid algorithm for large scale multiple traveling salesman problems. Expert Systems with Applications 139:112867.
- [11] Karapetyan D., Gutin G., 2011, Lin-Kernighan heuristic adaptations for the generalized traveling salesman problem. European Journal of Operational Research 208:221-232.

- [12] Karapetyan D., Gutin G., 2012, Efficient local search algorithms for known and new neighborhoods for the generalized traveling salesman problem. European Journal of Operational Research 219:234–251.
- [13] Khachai M.Y., Neznakhina E.D., 2017, Approximation schemes for the generalized traveling salesman problem. Proceedings of the Steklov Institute of Mathematics 299:97–105.
- [14] Laporte G., Mercure H., Nobert Y., 1987, Generalized traveling salesman problem through n sets of nodes: The asymmetric case. Discrete Applied Mathematics 18:185–197.
- [15] Lawrence V.S., Daskin M.S., 2006, A random-key genetic algorithm for the generalized traveling salesman problem. European Journal of Operational Research 174:38–53.
- [16] Lien Y., Ma E., Wah B.W.S., 1993, Transformation of the generalized traveling salesman problem into the standard traveling salesman problem. Information Sciences 74:177-189.
- [17] Lin S., Kernighan B.W., 1973. An effective heuristic Algorithm for the traveling-salesman problem. Operations Research 21:498-516.
- [18] Makarovskikh T., Panyukov A., Savitsky E., 2019, Software development for cutting tool routing problems. Procedia Manufacturing 29:567-574.
- [19] Mestria M., 2018, New hybrid heuristic algorithm for the clustered traveling salesman problem. Computers & Industrial Engineering 116:1–12.
- [20] Noon C., Bean J.C., 1991, A Lagrangian based approach for the asymmetric generalized traveling salesman problem. Operations Research 39(4):623-632.
- [21] Petunin A.A., Polishchuk E.G., Ukolov S.S., 2019, On the new algorithm for solving continuous cutting problems. IFAC 52(13):2320-2325
- [22] Renaud J., Boctor F.F., Laporte G., 1996, A fast composite heuristic for the symmetric traveling salesman problem. INFORMS Journal on Computing 8(2):134-143.
- [23] Renaud J., Boctor F.F., 1998, An efficient composite heuristic for the symmetric generalized travelling salesman problem. European Journal of Operational Research 108(3):571-584.

- [24] Salman R., Ekstedt F., Damaschke P., 2020, Branch-and-bound for the precedence constrained generalized traveling salesman problem, Operations Research Letters 48(2):163-166.
- [25] Smith S.L., Imeson F., 2017, GLNS: An effective large neighborhood search heuristic for the generalized traveling salesman problem. Computers & Operations Research 87:1–19.
- [26] Srivastava S.S., Kumar S., Garg R., Sen P., 1969, Generalized traveling salesman problem through n sets of nodes. CORS Journal 7:97–101.
- [27] Sundar K., Rathinam S., 2016, Generalized multiple depot traveling salesmen problem-Polyhedral study and exact algorithm. Computers & Operations Research 70:39-55.
- [28] Yang J., Shi X., Marchese M., Liang Y., 2008, An ant colony optimization method for generalized TSP problem. Progress in Natural Science 18:1417-1422.
- [29] Yuan Y., Cattaruzza D., Ogier M., Semet F., 2020, A branch-andcut algorithm for the generalized traveling salesman problem with time windows, European Journal of Operational Research 286(3):849-866.
- [30] Zhou X., Rodrigues B., 2017, An extension of the Christofides heuristic for the generalized multiple depot multiple traveling salesmen problem, European Journal of Operational Research 257(3):735-745.

Appendix A. Detailed Computational Results

The results of BS·, $\cdot = 0, \ldots, 4$ are reported in Tables A.1-A.5. The first column indicates the '.gts' label of the instance whereas the second column reports its best known solution z^{lit} , available in the literature. The next six triplets of columns report the BS. solution value z_{ω}^{BS} , $\cdot = 0, \ldots, 4$, its percent optimality gap $\Delta_{\omega}^{\cdot} = 100\% \frac{z_{\omega}^{-} - z^{lit}}{z^{lit}}$, and its runtime t_{ω}^{\cdot} in seconds when the beam width $\omega = 1, 2, 3, 4, 5, 10$.

		t_{10}^{0}	0.016	0.062	0.094	0.078	0.203	0.203	0.203	0.203	0.203	0.203	0.203	0.203	0.209	0.250	0.375	0.391	0.516	0.594	0.688	0.691	0.703	0.766	1.797	1.906	1.984	2.000	3.063	3.109	5.610	5.656	9.110	11.640	28.328	33.375	40.094	41.953		0.016	0.641	5.463	41.953
	$\omega = 10$	Δ_{10}^{0}	9.20	6.96	8.13	9.60	11.79	8.17	22.11	11.44	10.07	6.24	5.86	7.23	3.10	1.65	7.44	10.94	14.49	7.89	17.06	19.72	4.58	29.94	20.37	24.64	28.45	33.52	11.95	15.33	23.79	12.77	38.89	19.08	24.49	12.15	21.06	28.53		1.65	12.05	15.24	38.89
		z10	190	338	226	71159	10856	11172	11666	10531	10482	528	3864	267	8468	28357	39327	80340	48740	49508	12898	14601	53940	29450	1028	13158	17220	17506	76504	73822	1254	33321	31410	24728	7919	10824	72757	27835					
	-	t_5^0	0.016	0.031	0.047	0.047	0.125	0.125	0.109	0.094	0.109	0.110	0.110	0.125	0.125	0.140	0.219	0.219	0.297	0.328	0.325	0.342	0.361	0.375	0.812	0.893	0.982	1.000	1.212	1.342	2.266	2.283	3.626	4.940	12.891	15.641	18.921	20.922		0.016	0.327	2.542	20.922
	$\omega = 5$	Δ_5^0	9.20	7.91	6.70	18.69	11.79	19.18	29.35	9.69	9.97	6.04	7.45	7.23	4.53	1.65	6.75	15.19	18.32	7.68	15.52	23.42	4.58	28.39	20.49	28.54	21.24	36.12	12.31	15.69	29.32	12.37	38.99	17.78	27.48	13.47	21.58	29.21		1.65	15.36	16.50	38.99
		0 g %	190	341	223	77062	10856	12309	12358	10366	10472	527	3922	267	8585	28357	39076	33415	50370	19411	12728	15052	53940	29099	1029	13570	16254	17847	76752	74047	1310	33204	31433	24456	8109	10951	73067	27984					
		t_4^0	0.016	0.031	0.032	0.031	0.093	0.093	0.094	0.094	0.078	0.078	0.078	0.094	0.094	0.094	0.187	0.203 8	0.205	0.265	0.250	0.250	0.275	0.312	0.617	0.620	0.684	0.691	01.109	1.061	2.113	2.215	3.215	4.418	1.187	3.131	5.844	6.432		0.016	0.250	2.119	6.432
	i = 4	Δ_4^0	9.77 (0.76 (5.22	5.55 (1.79 (1.14 (2.29 (9.69	2.11 (5.04 (2.85 (9.64 (3.10 (1.65 (9.40 (5.97 (8.25	9.23 (2.59 (3.42 (4.50 (2.04 (0.49 (8.38	5.96 (5.80 (2.17	5.70	9.32	3.63	8.14	1.60	9.56 1	7.20 1:	5.32 14	9.35 10		1.65 (5.84	7.27	8.14 10
	3	z_{Φ}^{0}	191	350 10	222	1513 2	0856 1	2511 2	2639 33	366 9	0676 1:	527	1119 11	273	3468 :	3357	047	3981 13	0337 18	0121	2405 1:	5052 2:	, 6688	7659 2:	1029 20	2497 18	7020 20	7805 31	3660 1:	1053 1	1310 29	3577 1:	1240 33	5250 2	3241 29	1311 1/	5318 23	3014 29			Ŧ	Ξ.	ñ
		t_3^0	000	016	031	031 81	047 10	047 12	062 12	062 10	062 10	.063	.062	0.79	3 078	078 28	110 40	.156 85	188 50	209 50	234 12	250 15	250 55	297 27	578	609 12	625 17	625 17	938 76	.954 74	.703	702 35	734 31	453 25	360 8	859 11	860 75	437 28		000	222	.635	437
sults	=3	V 0 30	77 0.	18 0.	05 0.	42 0.	01 0.	18 0.	44 0.	69 0.	97 0.	63 0.	88 0.	64 0.	57 0.	65 0.	28 0.	28 0.	32 0.	78 0.	59 0.	42 0.	50 0.	95 0.	82 0.	60 0.	53 0.	11 0.	83 0.	77 0.	98 1.	28 1.	01 2.	12 3.	66 8.	03 9.	89 11.	17 12.	ummary	65 0.	65 0.	25 1.	01 12.
$3S_0 Re$	3	V 00	1 9.7	5 9.	0 10.0	9 21.	9 I5.(9 19.	8 31.	6 9.0	2 9.9	5.0	4 8.8	3 9.0	6 3.1	7 1.0	1 7.	6 16.5	9 13.:	4 9.	5 12.3	2 23.	9 4.	1 20.9	6 24.8	6 19.0	9 30.1	3 34.	8 14.8	3 16.'	7 31.5	9 14.5	8 39.0	8 22.	7 26.0	1 18.0	4 23.8	7 31.	ŝ	1.0	15.0	17.:	39.(
A.1: F		N	19	34	23	7882	1116	1230	1255	1036	1047	52	397	27	850	2835	3927	8420	4823	5037	1240	1505	5389	2741	106	1262	1749	1758	7847	7474	133	3376	3143	2535	805	1139	7445	2840					
Table		t_{2}^{0}	0.000	0.000	0.015	0.015	0.031	0.045	0.031	0.031	0.047	0.047	0.031	0.047	0.046	0.062	0.047	0.780	0.094	0.109	0.110	0.109	0.110	0.125	0.250	0.263	0.318	0.331	0.518	0.572	1.121	1.113	1.824	2.218	5.524	6.583	7.657	8.287		0.000	0.110	1.070	8.287
	$\omega = 2$	$\overline{\Delta}_{0}^{0}$	8.62	9.49	22.49	21.54	11.33	10.36	18.84	9.69	9.94	14.29	9.40	10.84	6.55	1.65	7.67	16.61	13.32	8.68	15.52	30.70	7.73	21.02	27.17	22.58	27.91	35.89	13.59	9.99	31.49	13.63	35.27	22.21	29.63	13.55	24.12	28.25		1.65	13.96	17.27	35.89
		2 ⁰ 2	189	346	256	78910	10811	11398	11354	10366	10470	568	3993	276	8751	28357	39413	84450	48239	49868	12728	15940	55564	27428	1086	12941	17148	17817	77626	70401	1332	33578	30592	25377	8246	10959	74593	27775					
		t_1^0	0.000	0.000	0.015	0.016	0.016	0.031	0.031	0.016	0.031	0.031	0.016	0.031	0.031	0.032	0.031	0.062	0.078	0.094	0.109	0.110	0.109	0.125	0.234	0.250	0.250	0.265	0.391	0.375	0.625	0.625	0.984	1.204	2.891	3.391	4.078	4.265		0.000	0.102	0.579	4.265
	$\omega = 1$	Δ_1^0	9.77	10.76	24.40	21.85	15.01	7.82	23.94	9.84	9.93	22.74	15.10	17.67	7.49	3.24	9.27	17.73	19.86	4.40	18.81	13.03	10.23	20.45	33.72	16.86	28.48	42.90	11.78	8.18	33.17	17.29	29.42	24.77	32.20	10.41	18.14	32.83		3.24	17.48	18.15	42.90
		z ⁰	191	350	260	79114	11169	11136	11841	10380	10469	610	4201	293	8828	28803	39997	85261	51024	47906	13090	13785	56853	27299	1142	12337	17224	18736	76393	69241	1349	34657	29268	25908	8409	10656	71001	28768					
		z^{lit}	174	316	209	64925	9711	10328	9554	9450	9523	497	3650	249	8213	27898	36605	72418	42570	45886	11018	12196	51576	22664	854	10557	13406	13111	68340	64007	1013	29549	22615	20765	6361	9651	60099	21657					
		Instance	11eil51	14st70	16ei176	16 pr 76	20 kroA100	$20 \mathrm{kroB100}$	20 kroC100	$20 \mathrm{kro} \mathrm{D100}$	$20 \mathrm{kroE100}$	20 rat 99	20 r d100	21eil101	21 lin 105	22 pr 107	25 pr 124	26 bier 127	28 pr 136	29 pr144	30 kroA150	30 kroB150	$31 \mathrm{pr} 152$	32u159	$39 \operatorname{rat} 195$	40d198	$40 \mathrm{kroA200}$	$40 \mathrm{kroB200}$	45ts225	$46 \mathrm{pr} 226$	53gil 262	53 pr 264	$60 \mathrm{pr} 299$	64 lin 318	80 rd400	84fi417	88 pr 439	89pcb442		Minimum	Median	Average	Maximum

		t_{10}^{1}	0.031	0.062	0.094	0.078	0.203	0.203	0.203	0.203	0.203	0.125	0.203	0.209	0.218	0.250	0.375	0.391	0.516	0.609	0.703	0.703	0.765	0.797	1.812	1.922	2.000	2.000	3.078	3.114	5.658	5.671	9.187	11.731	28.438	34.297	40.113	41.953		0.031	0.656	5.503	41.903
	$\omega = 10$	Δ^1_{10}	2.87	4.75	1.91	8.57	5.39	0.00	12.37	6.14	4.75	4.43	4.22	2.41	0.55	1.16	5.27	9.31	8.65	6.31	9.65	12.48	2.43	24.02	16.16	22.73	24.98	29.80	7.82	14.13	19.25	8.23	33.95	15.13	20.25	11.38	17.67	24.63		0.00	8.61	11.22	33.90
		z_{10}^{1}	179	331	213	70491	10234	10328	10736	10030	9975	519	3804	255	8258	28223	38535	79157	46252	48783	12081	13718	52830	28109	992	12957	16755	17018	73686	73050	1208	31981	30293	23907	7649	10749	70716	26992					
		$t_{\rm E}^1$	0.016	0.031	0.047	0.047	0.125	0.109	0.109	0.109	0.109	0.110	0.110	0.125	0.140	0.140	0.219	0.235	0.297	0.344	0.375	0.390	0.406	0.438	0.921	0.969	1.000	1.000	1.500	1.562	2.765	2.797	4.500	5.687	3.937	6.359	9.672	0.656		0.016	0.360	2.704	0.000
	v = 5	Δ_5^1	2.87	4.75	4.31	2.71	5.39	9.72	1.26	6.05	3.77	4.43	4.77	2.41	1.72	1.16	6.26	3.91	0.64	6.10	9.33	9.29	2.43	3.72	5.69	5.09	6.30	2.76	8.17	4.07	2.21	8.86	3.47	5.25	2.48 1	2.68 1	8.32 1	4.95 2		1.16	0.18	2.43 0.47 0	3.41 4
	2	21 22	179	331	218	3175 1	0234	1332	1585 2	0022	9882	519	3824	255	8354	8223	8897	2492 1	7099 1	8686	2046	4548 1	2830	8041 2	988 1	3206 2	5591 1	7406 3	3924	3013 1	1238 2	2166	0185 3	3931 1	7791 2	0875 1	1110 1	7061 2			-	- 0	o
		t_4^1	.016	.031	.031	.031 7	.094 1	.094 1	.094 1	.094 1	.093	.079	.078	.094	.109	.125 2	.188 3	.187 8	.266 4	.266 4	.313 1	.312 1	.344 5	.375 2	.734	.781 1	.828 1	.828 1	.234 7	.266 7	.234	.250 3	.593 3	.578 2	.234	.203 1	.875 7	.672 2		.016	.289	.184	710
	=4	Δ^1_4	.17 0	.27 0	.91 0	.36 0	.39 0	.78 0	.19 0	.05 0	.47 0	.43 0	.95 0	.02 0	.55 0	.16 0	.94 0	.14 0	.19 0	.54 0	.32 0	.29 0	.58 0	.69 0	.69 0	0 00.	.94 0	.62 0	.73 1	.06 1	.00 2	.45 2	.20 3	.98 4	.55 11	.10 13	.37 15	.23 16		.55 0	.01 0	.40 2	0T 07.
	Э	z41	83 5.	20 1.	13 1.	43 20.	34 5.	61 16.	56 26.	22 6.	53 3.	19 4.	86 11.	64 6.	58 0.	23 1.	178 8.	85 15.	84 9.	89 6.	25 7.	48 19.	07 2.	73 17.	88 15.	52 16.	81 22.	57 31.	22 7.	09 14.	46 23.	37 10.	76 35.	06 18.	86 25.	0.5 16.	42 22	21 25.		0	13.	13. 25	.00
		-16	1	16 3	31 2	31 781	33 102	51 120	33 120	32 100	32 98	32	32 40	28	78 82	78 282	41 398	56 833	88 464	18 488	35 118	50 145	36 529	97 266	78	122	25 164	25 172	53 736	84 730	03 12	03 326	50 305	78 247	75 79	75 112	54 735	14 271		9	34		14
ılts		+	0.01	0.01	0.0	0.0	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.07	0.07	0.07	0.14	0.15	0.15	0.21	0.26	0.25	0.26	0.25	0.57	0.61	0.62	0.62	36.0	0.98	1.70	1.70	2.75	3.47	8.37	9.87	11.95	12.64	nmary	0.01	0.25	12.65	12.04
3 ₁ Rest	$\omega = 3$	Δ_3^1	5.17	6.33	4.31	20.37	10.93	16.78	24.41	6.05	3.77	4.43	2.27	6.02	1.35	1.16	5.51	10.51	2.82	7.73	7.32	19.29	2.58	14.49	19.20	17.20	24.35	26.25	9.94	14.92	27.34	11.26	33.53	17.27	22.81	17.19	19.61	25.22	Sun	1.16	11.10	13.05	33. 00
A.2: BS		-16 2 2 3	183	336	218	78152	10772	12061	11886	10022	9882	519	3733	264	8324	28223	38623	80032	43772	49431	11825	14548	52907	25948	1018	12373	16670	16553	75136	73558	1290	32877	30197	24352	7812	11310	71885	27119					
Table /		t_{2}^{1}	0.000	0.015	0.015	0.015	0.047	0.046	0.047	0.031	0.047	0.016	0.047	0.047	0.047	0.062	0.093	0.109	0.156	0.172	0.188	0.188	0.188	0.219	0.422	0.438	0.438	0.437	0.656	0.656	1.156	1.157	1.843	2.328	5.610	6.625	7.985	8.328		0.000	0.180	1.108	0.320
	$\omega = 2$	Δ_2^1	6.90	1.27	17.22	19.13	5.39	0.23	11.25	6.05	2.60	6.44	7.56	2.41	1.75	1.16	4.55	15.53	2.82	5.95	9.33	27.71	2.44	17.69	18.15	20.26	22.09	30.97	8.74	8.89	27.44	10.70	31.27	16.86	25.06	12.47	17.98	22.24		0.23	10.02	12.46	31.27
		1 ⁶ 2	186	320	245	77347	10234	10352	10629	10022	9771	529	3926	255	8357	28223	38271	83663	43772	48616	12046	15576	52835	26673	1009	12696	16367	17172	74311	669690	1291	32711	29687	24267	7955	10854	70902	26474					
		t_1^1	0.016	0.015	0.016	0.016	0.016	0.032	0.031	0.016	0.031	0.032	0.016	0.031	0.032	0.032	0.031	0.063	0.094	0.094	0.110	0.110	0.125	0.140	0.235	0.250	0.281	0.331	0.438	0.485	0.687	0.641	1.015	1.219	2.906	3.516	4.306	5.016		0.015	0.102	0.623	010.6
	$\omega = 1$	Δ_1^1	4.02	3.16	19.14	18.10	10.93	0.68	15.41	6.21	2.59	14.08	13.56	13.25	1.00	1.16	6.08	16.65	13.85	2.23	6.70	10.13	3.84	12.97	22.72	14.04	22.46	26.25	6.72	7.26	22.21	10.70	20.87	18.74	26.80	9.39	12.25	23.78		0.68	12.61	12.22	20.8U
		-1 ×	181	326	249	76678	10772	10398	11026	10037	9770	567	4145	282	8295	28222	38832	84474	48468	46907	11756	13431	53558	25603	1048	12039	16417	16553	72930	68654	1238	32711	27335	24657	8066	10557	67460	26806					
		z^{lit}	174	316	209	64925	9711	10328	9554	9450	9523	497	3650	249	8213	27898	36605	72418	42570	45886	11018	12196	51576	22664	854	10557	13406	13111	68340	64007	1013	29549	22615	20765	6361	9651	66009	21657					-
		Instance	11eil51	14st70	16eil76	16 pr 76	$20 \mathrm{kroA100}$	$20 \mathrm{kroB100}$	20 kroC100	$20 \mathrm{kroD100}$	$20 \mathrm{kroE100}$	20 rat 99	20 rd100	21eil101	21 lin 105	22 pr 107	25 pr 124	26 bier 127	28 pr 136	$29 \mathrm{pr}144$	$30 \mathrm{kroA150}$	30 kroB150	31 pr 152	32° u159	$39 \operatorname{rat} 195$	40d198	$40 \mathrm{kroA200}$	$40 \mathrm{kroB200}$	45ts225	46 pr 226	53gil262	53 pr 264	60 pr 299	64 lin 318	80 r d400	84fi417	88 pr 439	89pcb442		Minimum	Median	Average	Maximum

		t_{10}^{2}	0.031	0.063	0.094	0.094	0.203	0.219	0.204	0.203	0.219	0.203	0.203	0.218	0.219	0.250	0.375	0.406	0.516	0.618	0.734	0.719	0.765	0.875	1.859	1.922	2.047	2.063	3.141	3.187	5.781	5.781	9.390	11.985	29.219	34.703	41.485	42.281		0.031	0.669	5.619	42.281
	$\omega = 10$	Δ^2_{10}	2.30	1.27	0.00	1.20	3.62	0.00	2.55	6.14	0.56	1.81	4.22	2.41	0.47	0.56	2.71	4.06	5.19	1.08	2.49	4.20	2.43	1.13	14.52	1.82	15.28	10.88	1.14	5.72	11.94	8.44	6.66	13.30	14.78	2.34	8.59	11.69		0.00	2.63	4.93	15.28
		z_{10}^{2}	178	320	209	65702	10063	10328	9798	10030	9576	506	3804	255	8252	28053	37598	75360	44778	46382	11292	12708	52830	22919	978	10749	15455	14537	69120	67667	1134	32044	24122	23526	7301	9877	65260	24188					
		t_5^2	0.015	0.031	0.047	0.032	0.125	0.109	0.110	0.110	0.109	0.109	0.110	0.125	0.125	0.140	0.218	0.218	0.282	0.328	0.390	0.391	0.422	0.453	0.937	1.000	1.032	1.031	1.563	1.609	2.906	2.875	4.688	5.623	4.360	7.907	1.484	2.703		0.015	0.359	2.881	2.703
	=5	$\Delta_{\rm E}^2$.30	.27	.39	.53	3.62	.46	20	.05	2.59	.81	1.77	20	.72	.56	.84	.50	.01	.98	3.49	8.72	2.43	6.13	1.17	.42	8.79	1.44	8.17	5.72	2.73	.53	.93	.04	3.19 I.	.1 66.	.36 2	0.74 2:		.46	5.24		3.19 2 ²
	3	215 215	178 2	320 1	214 2	163 6	063	376 (369 1	022 6	2 042	506 1	324 4	252 1	354 1)53 (110	295 5	553 7	714 3	±02	260 8	330 2	15 15 15	975 14	107	584 8	04 14	924 8	367 5	12 12	3 120	182 6	349 10	391 16	343 1	324 10	200					16
		0.4	.0	-	-	1 691	4 100	4 105	36	4 100	4 97	4	8 8		8000	5 28(7 38(7 792	0 455	1 477	2 114	8 132	0 528	5 26(0	3 107	0 145	4 150	5 739	1 676	1	32(4 24]	1 228	5 75	7 98	8 663	8 237	-	0	2		<u>∞</u>
		t.	0.00	0.03	0.03	0.03	0.09	0.09	0.09	0.09	0.09	0.09	0.07	0.09	0.10	0.12	0.18	0.18	0.25	0.28	0.31	0.32	0.36	0.37	0.75	0.81	0.86	0.84	1.26	1.28	2.31	2.32	3.73	4.78	11.48	14.90	17.95	17.68		0.00	0.29	2.34	17.95
	$\omega = 4$	Δ^2_{4}	0.00	1.27	0.00	5.21	3.62	2.07	0.00	6.05	1.92	1.81	10.88	5.22	0.55	0.56	2.56	9.50	6.25	3.57	7.21	8.72	2.58	7.47	14.17	4.84	15.31	12.94	7.73	5.33	7.60	8.35	12.21	11.06	16.43	5.74	8.42	10.02		0.00	5.90	6.31	16.43
		z42	174	320	209	68310	10063	10542	9554	10022	9706	506	4047	262	8258	28053	37542	79295	45231	47525	11812	13260	52907	24356	975	11068	15458	14808	73622	67421	1090	32016	25377	23061	7406	10205	65158	23827					
		t_{3}^{2}	0.015	0.016	0.032	0.031	0.078	0.062	0.063	0.078	0.079	0.063	0.063	0.078	0.078	0.094	0.156	0.157	0.188	0.235	0.250	0.266	0.281	0.313	0.578	0.625	0.657	0.656	0.969	0.984	1.765	1.750	2.844	3.625	8.625	10.543	12.469	13.156	ury -	0.015	0.243	1.720	13.156
Results	$\omega = 3$	∆ 235	0.00	1.27	2.39	0.11	0.01	0.46	5.17	6.05	2.59	1.81	2.25	3.21	1.33	0.56	2.34	9.88	2.82	3.87	7.21	8.72	2.58	6.36	7.03	7.27	12.07	9.88	9.94	5.72	11.85	9.16	9.68	8.87	12.29	5.83	13.61	11.90	Summa	0.00	5.78	5.72	13.61
3: BS ₂		C1 07 X	174	320	214	64994	9712	10376	10048	10022	0170	506	3732	257	8322	28053	37462	79574	43772	47660	11812	13260	52907	24105	914	11324	15024	14406	75136	67667	1133	32256	24804	22607	7143	10214	68278	24235					
ble A.		t_{2}^{2}	.000	.000	.015	.016	.047	.047	.046	.047	.047	.047	.047	.047	.062	.063	.110	.094	.156	.156	.203	.203	.219	.219	.422	.437	.453	.453	.672	.688	.187	.204	.922	.438	.813	.141	.203	.625		000.	.180	.154	.625
Ţ	=2	Δ_{2}^{2}	0.57 0	.27 0	3.83 0	1.38 0	3.62 0	0.23 0	0.37 0	3.05 O	2.60 0	3.44 0	.56 0	.20 0	75 0	.56 0	2.34 0	.98 0	.62 0	3.57 0	3.49 0	3.93 0	2.18 0	.47 0	0.95 0	1.77 0	.78 0	2.76 0	5.77 0	00.00	0.87 1	3.60 1	6.39 1	6.39 2	6.47 5	.50 7	.07 8	8 69.		00.00	0.08 0	6.28	3.93 8
	3	5 ² 5	75 C	20 1	17 3	66 4	63	52 C	49 5	22 6	71 2	29 6	26 7	52	57 1	53 C	62	95 11	59 1	25 3	02 3	05 18	00	56 7	39 6	61 4	89 17	84 12	82	07 C	13 5	89 89	95 15	85	45 15	06 C	48 7	88 11		0	ш	9	18
			-	3	5	677	100	5 103	5 104	5 100	5 97	- -	39	5	83	1 280	7 374	7 810	3 432	1 475	9 114	9 145	5 527	243	6	3 110	3 157	3 147	5 722	5 640	11	320	260	218	1 73	96 6	2 643	3 241			~1	~	
		t_1^2	0.00	0.00	0.016	0.016	0.032	0.016	0.015	0.015	0.015	0.031	0.016	0.031	0.031	0.031	0.047	0.047	370.0	0.094	0.109	0.10	0.125	0.141	0.250	0.266	0.260	0.266	0.375	0.375	0.594	0.641	1.00(1.281	2.984	3.609	4.175	4.406		0.000	0.102	0.597	4.406
	${\mathfrak{s} \atop = 1}$	Δ^2_1	0.57	1.27	6.70	5.05	0.01	0.68	6.65	5.42	2.59	4.02	4.79	13.25	1.00	0.54	4.77	11.10	12.73	0.22	5.12	4.58	3.44	2.78	10.77	9.11	13.85	9.02	5.67	0.09	11.25	9.05	4.27	8.41	13.66	1.42	8.24	8.19		0.01	5.09	5.84	13.85
		2 ⁻ 27	175	320	223	68205	9712	10398	10189	9962	9770	517	3825	282	8295	28050	38350	80458	47990	45988	11582	12754	53350	23295	946	11519	15263	14294	72215	64062	1127	32224	23580	22511	7230	9788	65050	23431					
		z^{lit}	174	316	209	64925	9711	10328	9554	9450	9523	497	3650	249	8213	27898	36605	72418	42570	45886	11018	12196	51576	22664	854	10557	13406	13111	68340	64007	1013	29549	22615	20765	6361	9651	60099	21657		u	-	e	m
		Instance	11eil51	14st70	16eil76	16 pr 76	20 kroA100	$20 \mathrm{kroB100}$	20 kroC100	$20 \mathrm{kroD100}$	$20 \mathrm{kroE100}$	20 rat 99	20 rd100	21 eil101	21 lin 105	22 pr 107	25 pr 124	26 bier 127	28 pr136	$29 \mathrm{pr}144$	$30 \mathrm{kroA150}$	30 kroB150	31 pr 152	32u159	39 rat 195	40d198	$40 \mathrm{kroA200}$	$40 \mathrm{kroB200}$	45ts225	46 pr 226	53gil 262	53 pr 264	60 pr 299	64 lin 318	80 rd400	84fi417	88 pr 439	89 pcb442		Minimu	Mediar	Averag	Maximu

		t_{10}^{3}	0.031	0.063	0.094	0.094	0.218	0.204	0.219	0.203	0.203	0.203	0.203	0.210	0.250	0.266	0.375	0.390	0.562	0.625	0.734	0.763	0.781	0.875	1.859	1.937	2.031	2.031	3.125	3.172	5.719	5.750	9.219	11.750	29.563	34.641	44.922	42.828		5.725	0.031	44.922	0.000
	$\omega = 10$	Δ^3_{10}	2.30	1.27	0.00	5.78	3.62	0.00	2.55	5.50	1.92	1.81	4.22	2.41	0.55	0.56	2.71	2.77	5.19	4.20	4.72	4.20	2.43	1.50	10.54	0.59	10.18	3.92	1.74	5.87	9.48	4.37	5.27	9.21	13.36	0.13	1.65	10.78		4.09	0.00	13.36	0.2.0
		z_{10}^{3}	178	320	209	68677	10063	10328	9798	9970	9706	506	3804	255	8258	28053	37598	74423	44778	47813	11538	12708	52830	23005	944	10619	14771	13625	69531	67767	1109	30840	23806	22678	7211	9664	61088	23992					
		$t_{5,3}^{t}$	0.015	0.015	0.047	0.046	0.109	0.109	0.109	0.109	0.125	0.109	0.125	0.125	0.141	0.156	0.235	0.221	0.297	0.391	0.438	0.391	0.422	0.469	0.937	0.983	1.031	1.031	1.563	1.610	2.375	2.875	4.688	5.844	4.438	7.500	1.312	1.156		2.821	0.015	1.312	U.391
	0 ≡5	Δ_{53}^3	2.30	1.27	2.39	3.63	3.62	0.46	1.20	5.42	2.59	1.81	4.77	2.41	1.72	0.56	3.84	6.91	6.94	3.99	3.49	2.90	2.43	1.40	0.07	2.37	8.79	5.57	2.09	0.09	8.29	2.26	4.39	6.04	0.41 1	0.82 1	7.07 2	5.47 2		4.16	0.09	1.40 2 2 55	o.oo
	3	ي تر م	178	320	214	7281	063	376	669	962	0220	506	824	255	3354	3053	8011	422	523	716	.402	2550	2830	5247 1	940 1	807	1584	841	0220	1062	760.	1216	3608	2020	023 1	1730	1345	841				1	
		t_{4}^{3}	16	16	32	32 67	93 10	94 10	94 6	94 6	94 6	78	94 5	60	3 60	00 28	72 38	87 73	50 45	81 47	83 11	43 12	50 52	75 25	20	12 10	44 14	90 13	32 66	97 64	13	97 30	35 25	82 22	47	41 5	12 64	41 25		49	16	41	22
			0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1(0.1(0.1(0.1'	0.18	0.2	0.2	0.25	0.3	0.3	0.3′	0.7	0.8	0.8	0.8	1.2	1.2	2.3	2.2	3.7:	4.78	11.5°	13.6°	16.3	17.1		2.2	0.0	17.1	Ň. D
	$\omega = 4$	Δ^3_{4}	0.00	1.27	0.00	0.93	3.62	2.07	0.00	5.42	1.92	1.81	4.55	5.22	0.55	0.56	2.56	1.36	6.25	3.57	7.21	1.39	2.58	7.47	10.07	0.54	9.54	1.64	1.65	0.09	8.29	1.89	3.09	9.52	9.67	0.67	6.14	5.74		3.58	0.00	10.07	70.7
		233 243	174	320	209	65526	10063	10542	9554	9962	9706	506	3816	262	8258	28053	37542	73402	45231	47525	11812	12366	52907	24356	940	10614	14685	13326	69467	64062	1097	30108	23313	22741	6976	9716	63791	22901					
s		t_{3}^{3}	0.015	0.016	0.031	0.032	0.062	0.063	0.078	0.078	0.062	0.062	0.063	0.078	0.078	0.094	0.156	0.172	0.234	0.234	0.265	0.266	0.281	0.297	0.594	0.625	0.688	0.688	0.925	1.047	1.875	1.766	2.781	3.531	8.656	10.203	12.266	13.000	ary	1.705	0.015	13.000	0.200
Result	$\omega = 3$	° ™	0.00	1.27	0.00	0.10	0.00	0.08	0.00	2.24	2.59	1.81	2.25	0.80	0.11	0.56	1.52	1.25	1.62	0.05	3.49	0.57	2.40	0.37	5.62	0.54	8.79	1.64	1.70	0.00	7.90	2.09	2.79	5.37	10.17	0.15	2.56	5.89	Summa	2.17	0.00	10.17	1.0/L
4: BS ₃		ი.ღ ჯ	174	320	209	64992	9711	10336	9554	9662	9770	506	3732	251	8222	28053	37162	73322	43259	45909	11402	12266	52814	22748	902	10614	14584	13326	69501	64007	1093	30166	23245	21880	7008	9665	61637	22932					
able A.		t_{2}^{3}	.000	0.016	0.016	0.032	.047	.047	.047	.047	0.046	.047	.047	0.063	0.062	0.062	.109	.109	0.156	.187	0.188	.203	.203	.219	.421	.453	.453	.453	.672	.688	.203	.218	.906	.391	.828	.860	.219	.578		.147	000.	.578	1 201.
T_{ϵ}	=2	Δ_{23}^{3}	.57 C	.27 0	.39 C	.38	.62 C	0.23 C	.37 0	.42 0	.60 C	.44 C	.56 C	.41 C	.75 C	56 C	.39 C	.41 C	.62 C	.57 C	.49 C	.74 C	0.18 C	.47 C	.85 C	.41 C	.91 C	.76 C	.41 C	000 C	.03	.14 1	.16 1	.68	.27 5	0.23 6	.56 8	.34 8		.87 1	00.0	0.91 8 67 9 67	ר ייסי
	3	n.č.	75 0	20 1	14 2	36 4	53 33	52 C	49 5	32 5	71 2	29 6	26 7	55	57 1	53 C	81 2	42 1	59 1	25 33	02	30	20	56 7	04	11 2	38 1C	29 7	00	07 C	34	82	34 7	37 4	87 8	73 0	35	31 6		ср	0	10	1
				8	5	6770	1000	103	104^{\prime}	66	.26	<u>с</u>	39:	10	83	280	3748	734	432	475:	114(125:	527(243	6	108	1480	141;	669	640(10	3018	242:	217	68	96	616	230:					
		t_{1}^{3}	0.00	0.016	0.016	0.016	0.016	0.031	0.031	0.032	0.031	0.016	0.031	0.031	0.032	0.032	0.047	0.047	0.078	0.094	0.110	0.110	0.125	0.141	0.250	0.266	0.250	0.297	0.391	0.375	0.656	0.672	1.078	1.297	3.000	3.515	4.218	4.391		0.604	0.000	4.391	70T.U
	$\frac{s}{1}$	Δ^3	0.57	1.27	6.70	5.05	0.01	0.68	6.65	5.42	2.59	4.02	4.79	8.84	1.00	0.54	4.77	6.05	10.97	0.22	5.12	5.06	3.44	0.37	5.62	5.41	9.44	5.29	3.76	0.00	6.22	1.97	3.97	6.94	6.82	1.16	4.83	6.04		4.21	0.00	10.97	4.01
		z.3	175	320	223	68205	9712	10398	10189	9962	9770	517	3825	271	8295	28050	38350	76799	47241	45988	11582	12813	53350	22747	902	11128	14672	13804	70909	64007	1076	30132	23512	22207	6795	9763	63001	22966					
		z^{lit}	174	316	209	64925	9711	10328	9554	9450	9523	497	3650	249	8213	27898	36605	72418	42570	45886	11018	12196	51576	22664	854	10557	13406	13111	68340	64007	1013	29549	22615	20765	6361	9651	60009	21657					
		Instance	11eil51	14st70	16eil76	16 pr 76	$20 \mathrm{kroA100}$	$20 \mathrm{kroB100}$	$20 \mathrm{kroC100}$	$20 \mathrm{kroD100}$	$20 \mathrm{kroE100}$	20 rat 99	20 r d100	21eil101	21lin105	22 pr 107	25 pr124	26 bier 127	28 pr136	$29 \mathrm{pr}144$	30 kroA150	30 kroB150	31 pr 152	32u159	$39 \operatorname{rat} 195$	40d198	$40 \mathrm{kroA200}$	$40 \mathrm{kroB200}$	45 ts 225	46 pr 226	53gil262	53 pr 264	60 pr 299	641in318	80 r d400	844417	88 pr 439	89pcb442		Average	Minimum	Maximum	Median

 0.0	1.625 0.687
3.18 0.00	1.06 8 2.45
	1.
 	4 0
3.383	28.95 0.400
3.07 0.00	10.19 2.62
3.106 0.015	1.953
00 83	19 24 05 (
N O	10.
2.006 0.031	15.312 0.266
$2.20 \\ 0.00$	7.03 1.90
.636	422
26 1 00 0	37 13 38 0
.0.0 .0.0	6.0
$0.784 \\ 0.000$	5.703 0.110
3.03 0.00	8.84 2.73
ц	ш
rage	ximu dian
	srage 3.03 0.784 3.26 1.636 2.20 2.006 1 imum 0.00 0.000 0.000 0.000 0.000 0.031