Appendix

A. Simpler Approaches That Do Not Work

The algorithm `ialg` is slow, in part, because it creates a branch for every subset of subtours (or half of them, anyway). Is this really necessary? Would it suffice to consider just one subtour constraint? Perhaps for a smallest subtour? Here, we show that simple heuristics like this can fail miserably. For concreteness, we propose the following approach that we call the `single_SEC_heuristic`().

`single_SEC_heuristic`():

- $\mathcal{S} \leftarrow \{\}$
- while True
    1. compute a solution to DFJ$(G, \mathcal{S})$
    2. if DFJ$(G, \mathcal{S}) = $ TSP(G), then return $|\mathcal{S}|$
    3. find the vertex sets $V_1, V_2, \dots, V_t$ of the DFJ solution over $\mathcal{S}$
    4. $\mathcal{S} \leftarrow \mathcal{S} \cup \{S\}$ where S is one of $V_1, V_2, \dots, V_t$

PROPOSITION 3. *For some classes of instances, just one SEC is required to prove the optimality of a TSP tour, but the* `single_SEC_heuristic`() *may add* $\Omega(2^n)$ *SECs. For example, this is true for Euclidean instances with* 3 *points located at the* x, y*-coordinate* $(0,0)$ *and* $n-3$ *points at* $(1,0)$.

Proof. For each integer n for which $q := (n-3)/2 \geq 7$ is odd, we construct such an instance. Let the vertex set be $V = V_0 \cup V_1$ where $V_0 = \{1,2,3\}$ are the vertices at $(0,0)$ and $V_1 = \{4,5,\dots,n\}$ are the vertices at $(1,0)$. An optimal tour is 1-2-3-$\cdots$-n-1 of weight 2, and the single SEC $x(E(S)) \leq |S|-1$ for $S = V_0$ suffices. Now, consider a cycle C_0 over V_0. For every $S \subseteq V_1$ with $2 < |S| < |V_1|/2 = (n-3)/2 = q$, consider a cycle C_1 over S and a cycle C_2 over $V_1 \setminus S$. Any 2-factor (C_0, C_1, C_2) constructed in this way has zero weight. If `single_SEC_heuristic` encountered them (in any arbitrary sequence), it would add a SEC for each S. The number of them is $(2^{2q} - 2 - 2(2q) - 2\binom{2q}{2})/2 = \Omega(2^{2q}) = \Omega(2^{2(n-3)/2}) = \Omega(2^n)$. □

Next, we consider a variant of the `single_SEC_heuristic` in which the vertex subset S in line 4 is always taken as a smallest vertex set among $V_1, V_2, \dots, V_t$. We call this variant the `smallest_SEC_heuristic`().

PROPOSITION 4. *For some classes of instances, just one SEC is required to prove optimality of a TSP tour, but the* `smallest_SEC_heuristic`() *may add* $\Omega(2^{2n/3}) = \Omega(1.5874^n)$ *SECs. For example, this is true for Euclidean instances with* q *points located at the* x, y*-coordinate* $(0,0)$ *and* $2q$ *points at* $(1,0)$.

Proof. For each odd $q \geq 7$, we construct such an instance. Let the vertex set be $V = V_0 \cup V_1$ where $V_0 = \{1,2,\dots,q\}$ are the vertices at $(0,0)$ and $V_1 = \{q+1, q+2, \dots, 3q\}$ are the vertices at $(0,1)$. An optimal tour is 1-2-3-$\cdots$-n-1 of weight 2, and the single SEC $x(E(S)) \leq |S|-1$ for $S = V_0$ suffices. Now, consider a cycle C_0 over V_0. For every $S \subseteq V_1$ with $2 < |S| < |V_1|/2 = q$, consider a cycle C_1 over S and a cycle C_2 over $V_1 \setminus S$. Any 2-factor (C_0, C_1, C_2) constructed in this way has zero weight. If `smallest_SEC_heuristic` encountered them (in any arbitrary sequence), it would add a SEC for each S. The number of them is $(2^{2q} - 2 - 2(2q) - 2\binom{2q}{2})/2 = \Omega(2^{2n/3})$. □

B. On a Different Question

What would happen if we posed the question differently? What if we sought the minimum number of SECs that necessarily *produces* a tour? Although this could be an interesting task for future work, one should not ignore the fact that essentially *all* SECs are necessary when the edges have the same cost. Indeed, this observation is what prompted us to phrase the question in terms of proving the optimality of a tour rather than producing it.

REMARK 1. Consider a TSP instance with $n \geq 5$ vertices in which all edges have the same cost. The number of SECs needed to necessarily *produce* a TSP tour is $2^{n-1} - \binom{n}{2} - n - 1 = \Theta(2^n)$, but zero suffice to prove optimality.

Proof. For every $S \subseteq V$ with $2 < |S| < n-2$, we must impose the SEC for either S or its complement $\overline{S} = V \setminus S$, because a cycle over S and a cycle over $\overline{S}$ will satisfy all other SECs and have minimum weight. The number of vertex subsets $S \subseteq V$ that satisfy $2 < |S| < n-2$ is $2^n - 2 - 2n - 2\binom{n}{2}$. Halving this gives the result. □

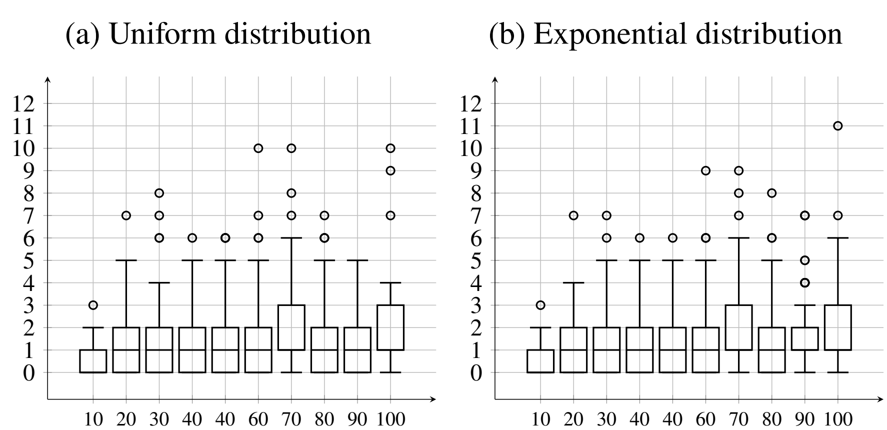
C. Randomly Generated Instances

We consider three classes of randomly generated instances. The first two have edge costs drawn at random, either from a uniform or exponential distribution. The third randomly places points in a square and uses Euclidean distances. For numerical reasons, we round costs to the nearest integer.

The first two classes are motivated by the work of Osorio and Pinto (2003) who claim that costs drawn from the uniform distribution lead to easier instances than from the exponential distribution. The way they measure difficulty is by the number of branch-and-bound nodes needed to prove optimality. The greater such number, the harder the instance. We sampled costs for instances with $n \in \{10, 20, \ldots, 100\}$ vertices. For the uniform distribution, we sample costs between 0 and 1000 and round to the nearest integer. For the exponential distribution, we set $\lambda = \frac{1}{999}$, as suggested by Osorio and Pinto to allow a higher probability of sampling numbers between 0 and 1000, and then round to the nearest integer. For each n, we sample 100 instances.

Results for the first two classes are given in Figure 4. For each each n, we give a boxplot showing the number of SECs required across the 100 samples. We observe that roughly 50% of the instances require 0 or 1 SEC, roughly 75% require at most two SECs, and no cases require more than 11 SECs. All in all, it seems that both of these classes are quite easy.

Figure 4 Results for randomly generated instances whose edge costs are drawn from the uniform and exponential distributions.



Finally, in the third class, we randomly place n points in the unit square and use Euclidean distances. To get a diverse set of edge costs, we multiply the distances by 1000 and round to the nearest integer. We find that computing the minimum number of SECs for these instances is much harder than before, so we sample 100 instances for each $n \in \{10, 15, \ldots, 50\}$.

Figure 5 Results for randomly generated Euclidean instances in the unit square.

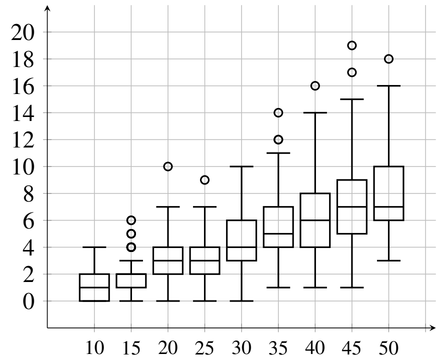
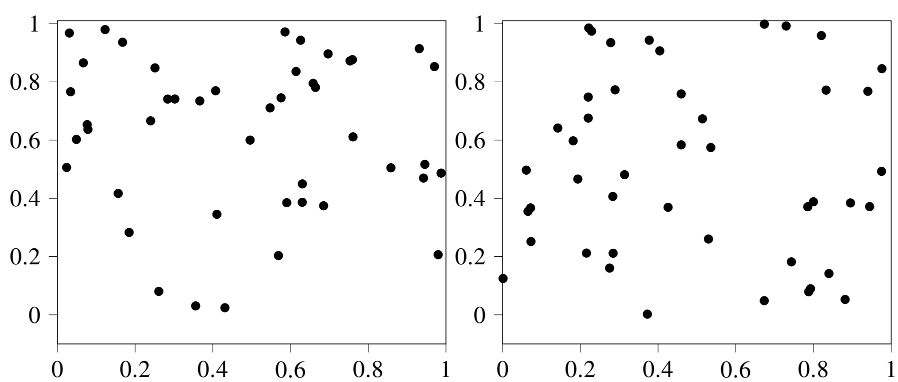


Figure 5 gives results. We observe that the number of SECs grows more quickly than before, already reaching a median of 7 SECs when $n = 45$, whereas only the outliers for the other two classes required this many—even when $n = 100$. We also see some $n = 45$ instances that require up to 19 SECs, while others require only one. Figure 6 shows one instance at each of these extremes.

Figure 6 Two instances sampled from the unit square. The left instance needs just one SEC, while the right needs 19 SECs.



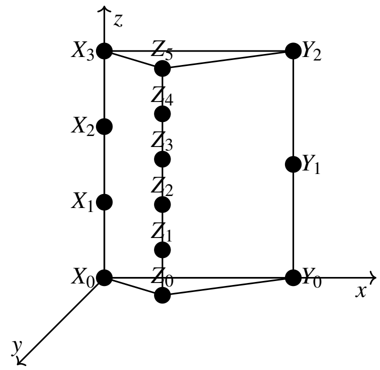
D. Rectilinear 3-Dimensional Instances

Previously, we have seen a class of TSP instances on n vertices that require a linear number $\Theta(n)$ of SECs (see Proposition 2) and several instaces requiring many SECs (see Tables 2 and 3). Here, we consider a class of instances that seem to require a quadratic number $\Theta(n^2)$ of SECs.

Specifically, Zhong (2021) proposes the *Rectilinear 3-Dimensional instances*, positioning n points on 3 parallel lines in $\mathbb{R}^3$, forming a prism in three-dimensional space. Following their notation, the coordinates of the vertices are given by $X_s = \left(0, 0, \frac{s}{i+1}\right)$, $s \in \{0, \ldots, i+1\}$, $Y_s = \left(\frac{1}{i+1} + \frac{1}{j+1}, 0, \frac{s}{j+1}\right)$, $s \in \{0, \ldots, j+1\}$ and $Z_s = \left(\frac{1}{i+1}, \frac{1}{k+1}, \frac{s}{k+1}\right)$, $s \in$

$\{0,\dots,k+1\}$, where X_0, Y_0, and Z_0 form the base of the prism. The positions of the n points depend on three parameters: i, j, k, such that $n = i + j + k + 6$, and the rectilinear distance between points is used. Figure 7 shows an example for $n = 13$.

Figure 7 An example of rectilinear 3-dimensional instance, with $n = 13$ **and hence** $i = 2, j = 1, k = 4$



Zhong shows that certain choices of i, j, k make these instances hard for Concorde. Here, we tune these parameters in an attempt to find instances that require $\Theta(n^2)$ SECs. In the case where $n \equiv 1 \pmod 3$, we set the parameters as follows $i = \frac{n-1}{3} - 2,\ j = i - 1,\ k = i + 2$. For convenience, we relabel the vertices into three groups, one for each parallel line:

$$V_1 = \{0, 1, \dots, i+1\}, \qquad V_2 = \{i+2, \dots, 2i+2\}, \qquad V_3 = \{2i+3, \dots, n-1\}.$$

For such instances, we have empirically found that the smallest collection of SECs needed to prove optimality often has the following *chain structure*: For each group of nodes V_j, we have all the chains of length k, for $k \in \{3, \dots, |V_j|\}$. Empirically, this holds for $n \in \{13, 16, 19, 22, 25, 28, 31, 34\}$. For $n \geq 37$, the `ialg` computation is too demanding; this is not surprising given that the Rectilinear 3-dimensional instances are arguably the hardest metric TSP instances in the literature.

Example. For $n = 13$, we have $i = 2$, and the three groups are:

$$V_1 = \{0, 1, 2, 3\} \qquad V_2 = \{4, 5, 6\} \qquad V_3 = \{7, 8, 9, 10, 11, 12\}.$$

In this case, the SECs we add are $x(E(S)) \leq |S| - 1$ for S being:

$$\{0,1,2\}, \{1,2,3\}, \{0,1,2,3\}, \{4,5,6\}, \{7,8,9\}, \{8,9,10\}, \{9,10,11\}, \{10,11,12\},$$
$$\{7,8,9,10\}, \{8,9,10,11\}, \{9,10,11,12\}, \{7,8,9,10,11\}, \{8,9,10,11,12\},$$
$$\{7,8,9,10,11,12\}.$$

Generally, the number of chain SECs can be computed as:

$$\sum_{t=1}^{i} t + \sum_{t=1}^{i-1} t + \sum_{t=1}^{i+2} t = \frac{3i^2 + 5i + 6}{2},$$

which grows quadratically in n. Proving whether these instances truly require a quadratic number of SECs is an interesting question for future work.