An FPT Algorithm for Splitting a Necklace Among Two Thieves

Michaela Borzechowski
Department of Mathematics and Computer Science, Freie Universität Berlin, Germany

Patrick Schnider
Department of Computer Science, ETH Zürich, Switzerland

Simon Weber
Department of Computer Science, ETH Zürich, Switzerland

Abstract

It is well-known that the 2-Thief-Necklace-Splitting problem reduces to the discrete Ham Sandwich problem. In fact, this reduction was crucial in the proof of the PPA-completeness of the Ham Sandwich problem [Filos-Ratsikas and Goldberg, STOC'19]. Recently, a variant of the Ham Sandwich problem called α-Ham Sandwich has been studied, in which the point sets are guaranteed to be well-separated [Steiger and Zhao, DCG'10]. The complexity of this search problem remains unknown, but it is known to lie in the complexity class UEOPL [Chiu, Choudhary and Mulzer, ICALP'20]. We define the analogue of this well-separation condition in the necklace splitting problem – a necklace is *n-separable*, if every subset A of the n types of jewels can be separated from the types $[n] \setminus A$ by at most n separator points. Since this version of necklace splitting reduces to α-Ham Sandwich in a solution-preserving way it follows that instances of this version always have unique solutions.

We furthermore provide two FPT algorithms: The first FPT algorithm solves 2-Thief-Necklace-Splitting on $(n-1+\ell)$-separable necklaces with n types of jewels and m total jewels in time $2^{O(\ell \log \ell)} + O(m^2)$. In particular, this shows that 2-Thief-Necklace-Splitting is polynomial-time solvable on n-separable necklaces. Thus, attempts to show hardness of α-Ham Sandwich through reduction from the 2-Thief-Necklace-Splitting problem cannot work. The second FPT algorithm tests $(n-1+\ell)$-separability of a given necklace with n types of jewels in time $2^{O(\ell^2)} \cdot n^4$. In particular, n-separability can thus be tested in polynomial time, even though testing well-separation of point sets is co-NP-complete [Bergold et al., SWAT'22].

2012 ACM Subject Classification Mathematics of computing → Combinatoric problems; Theory of computation → Computational geometry; Theory of computation → Fixed parameter tractability

Keywords and phrases Necklace splitting, n-separability, well-separation, ham sandwich, FPT

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2023.15

Related Version *Full Version*: https://arXiv.org/abs/2306.14508

Funding *Michaela Borzechowski*: DFG within the Research Training Group GRK 2434 *Facets of Complexity*.
Simon Weber: Swiss National Science Foundation under project no. 204320.

1 Introduction

The *necklace splitting problem* is one of the most famous problems in fair division. It is usually illustrated by the following story: two thieves have stolen a valuable necklaces with n different types of jewels (diamonds, rubies, etc.). They want to divide their bounty fairly between them, that is, in such a way that both of them get the same number of jewels of each type. As cutting through the necklace takes a lot of effort, they want to do this with as few cuts as possible. A mathematically inclined thief who knows the necklace splitting theorem [1, 3, 12] will realize that no matter how the jewels are ordered on the necklace, n cuts will always suffice for this. However, all known proofs of this result are of a topological nature and do

34th International Symposium on Algorithms and Computation (ISAAC 2023).
Editors: Satoru Iwata and Naonori Kakimura; Article No. 15; pp. 15:1–15:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

not give our thief any information on how to find the cuts. Thus, a more algorithmically inclined thief might wonder whether a set of n cuts can be found efficiently. Unfortunately, it turns out that the search problem of finding n cuts is in general PPA-complete [11], making an efficient algorithm unlikely. In this paper, we study separability conditions under which the thieves can find the cuts efficiently.

The ideas for the separability conditions stem from a variant of another famous fair division problem, namely the *Ham Sandwich problem*. The Ham Sandwich theorem [19] states that any d point sets (or mass distributions) in $\mathbb{R}^d$ can be simultaneously bisected by a single hyperplane. Again, finding such a Ham Sandwich Cut is in general PPA-complete [11]. However, under the assumption that the point sets are *well-separated* (which we will formally define in Section 2), the cut is unique [18] and the corresponding search problem lies in the complexity class UEOPL [5]. UEOPL is a subclass of PPA. It is conjectured to be a strict subclass, with a recent paper showing a black-box separation between the two classes [14].

The Ham Sandwich problem and the necklace splitting problem are intimately related. In fact, the necklace splitting theorem can be proved by lifting the necklace with n types of jewels to the *moment curve* in $\mathbb{R}^n$, which is the curve parameterized by $(t, t^2, t^3, \ldots, t^n)$, and then applying the Ham Sandwich theorem. By the same idea, the PPA-hardness for the Ham Sandwich problem follows from the PPA-hardness of the necklace splitting problem. In the well-separated setting, no hardness result is known for finding the now unique Ham Sandwich cut. A natural approach to show for example UEOPL-hardness of this problem would be to show hardness for a necklace splitting variant whose lifts give well-separated point sets. This leads to the definition of *n-separable necklaces*, which we again define formally in Section 2.

However, as we show in this paper, this approach will not work, as the necklace splitting problem on n-separable necklaces can be solved in polynomial time. Relaxing the notion of separability further, we get an FPT algorithm for the necklace splitting problem, parameterized by the separability:

▶ **Theorem 1.** *2-Thief-Necklace-Splitting can be solved in time $2^{O(\ell \log \ell)} + O(m^2)$ on every $(n-1+\ell)$-separable necklace C with n types of jewels and m total jewels.*

We also provide an FPT algorithm to check whether a necklace is $(n-1+\ell)$-separable. This is again in contrast to the Ham Sandwich problem, where it has been shown that checking well-separation is co-NP-complete [4].

Our work provides the first FPT viewpoint on the necklace splitting problem, which so far has only been studied from the viewpoint of approximation algorithms [2].

2 Preliminaries

2.1 Separability and Unique Solutions

▶ **Definition 2** (Necklace). *A* necklace *is a family C of disjoint finite point sets in $\mathbb{R}$. The sets in C are called* colors*.*

Note that in the literature, the points in each color $c \in C$ are also called *beads* or *jewels* of color c. Furthermore, this kind of necklace is sometimes also called an *open* necklace, since the colors are arranged in $\mathbb{R}$ and not on a cycle.

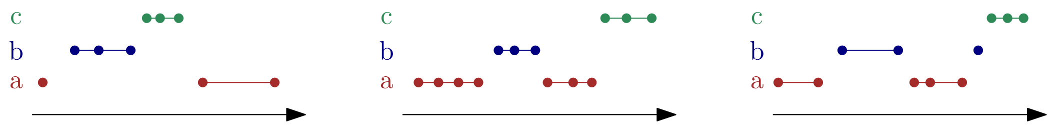
For simplicity, in the rest of this paper we assume that each color has an *odd* number of points. All of our results can be adapted to the more general setting without this restriction, or even to the setting where colors are finite unions of intervals. However, the definitions and proofs have to be adjusted carefully. We discuss these possible extensions of our results in the full version of the paper.

Since colors in a necklace are disjoint, we can view our necklace as a string over the alphabet C: each color defines one character and the sequence of characters is defined by the order in which the colors appear when going from $-\infty$ to ∞, with consecutive occurrences of the same color yielding just one character. See Figure 1 for an example.

We call the number of occurrences of a color c in this string the number of *components* it consists of. We say a color $c \in C$ is an *interval*, if it consists of exactly one component. In other words, a color c is an interval if its convex hull does not intersect any other color $c' \in C$. In Figure 1a, the green color c is an interval, whereas the red color a is not, it consists of two components.

▶ **Definition 3** (Separability). *A necklace C is k-*separable *if for all $A \subseteq C$ there exist k* separator points *$s_1 < \ldots < s_k \in \mathbb{R}$ that separate A from $C \setminus A$. More formally, if we alternatingly label the intervals $(-\infty, s_1], [s_1, s_2], \ldots, [s_{k-1}, s_k], [s_k, \infty)$ with A and $\overline{A}$, for every interval I labelled A we have $I \cap \bigcup_{c \in (C \setminus A)} c = \emptyset$ and for every interval I' labelled $\overline{A}$ we have $I' \cap \bigcup_{c \in A} c = \emptyset$.*

The separability *$sep(C)$ of a necklace C is the minimum integer $k \geq 0$ such that C is k-separable.*



(a) "$a\ b\ c\ a$" is 2-separable. **(b)** "$a\ b\ a\ c$" is 3-separable. **(c)** "$a\ b\ a\ b\ c$" is 4-separable.

Figure 1 Necklace C with 3 colors a, b and c.

Note that for a necklace with n colors, $sep(C) \geq n - 1$, and this is tight, as can be seen in Figure 1a. Our definition of k-separability is strongly related to the well known notion of *well-separation*.

▶ **Definition 4.** *Let $P_1, \ldots, P_k \subset \mathbb{R}^d$ be point sets. They are* well-separated *if and only if for every non-empty index set $I \subset [k]$, the convex hulls of the two disjoint subfamilies $\bigcup_{i \in I} P_i$ and $\bigcup_{i \in [k] \setminus I} P_i$ can be separated by a hyperplane.*

A set of two colors in $\mathbb{R}$ is 1-separable if and only if it is well-separated. Furthermore we observe the following property.

▶ **Lemma 5.** *Let C be a set of n colors in $\mathbb{R}$. Let C' be the set of subsets of $\mathbb{R}^n$ obtained by lifting each point in each color of C to the n-dimensional moment curve using the function $f(t) = (t, t^2, \ldots, t^n)$. Then the set C is n-separable if and only if C' is well-separable.*

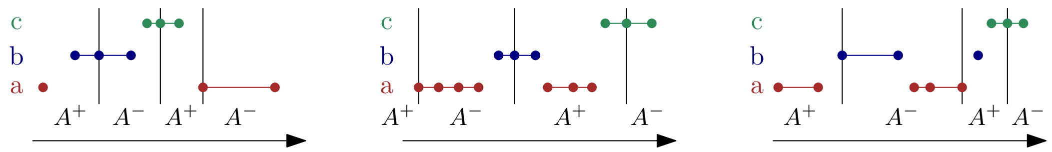
Proof. If C is n-separable, for each subset A of C, there exist n points $S = (s_1, \ldots, s_n)$ partitioning C into intervals alternatingly labelled A and $\overline{A}$. Let H be the hyperplane that goes through these separator points S lifted to the moment curve. By [15, Lemma 5.4.2], at each separating point, the moment curve passes from one side of H to the other. The points belonging to intervals labelled A lie on one side of the hyperplane and the points belonging to intervals labelled $\overline{A}$ lie on the other side. Since this holds for all subsets of C, it follows that C' is well-separated.

If C' is well-separated, for each subset A' of colors, there exists a hyperplane that separates A' from $C' \setminus A'$. By [15, Lemma 5.4.2], this hyperplane intersects the moment curve at at most n points. These points define the separator points that show that C is n-separable. ◀

The problem of *Necklace Splitting* is that two thieves want to split the necklace they stole into equal parts with as few cuts as possible. Mathematically we partition the necklace into several intervals which belong to each thief in turn.

▶ **Definition 6** (2-Thief-Necklace-Splitting). *Given a necklace C with n colors, find n* split points *that split the necklace into $n+1$ open intervals alternatingly labelled A^+ and A^-, such that for each color $c \in C$, the union of all intervals labelled A^+ contains the same number of points of c as the union of all intervals labelled A^-.*

It is well known that there always exists a solution to this problem [1, 3, 12]. Note that due to our assumption of every color containing an odd number of points, every solution must contain exactly one point per color as a split point.



(a) Solution for "$a\ b\ c\ a$". **(b)** Solution for "$a\ b\ a\ c$". **(c)** Solution for "$a\ b\ a\ b\ c$".

Figure 2 Example of solutions to 2-Thief-Necklace-Splitting.

▶ **Theorem 7.** *Let C be an n-separable necklace with n colors. There is a unique solution to 2-Thief-Necklace-Splitting on C.*

In order to prove the above theorem, we consider the classical reduction of 2-Thief-Necklace-Splitting to the Ham-Sandwich problem obtained by lifting the points to the moment curve, as it appeared in many works before [7, 11, 15, 17]. However, since the necklace we apply this reduction to is n-separable, by Lemma 5, the resulting points are well-separated, which allows us to apply the following stronger version of the Ham-Sandwich theorem due to Steiger and Zhao [18].

▶ **Lemma 8** (α-Ham-Sandwich Theorem, [18]). *Let $P_1, \ldots, P_n \subset \mathbb{R}^n$ be finite well-separated point sets in* weak general position[1]*, and let $\alpha_1, \ldots, \alpha_n$ be positive integers with $\alpha_i \leq |P_i|$, then there exists a unique $(\alpha_1, \ldots, \alpha_n)$-cut, i.e., a hyperplane H that contains a point from each color and such that for the closed positive halfspace H^+ bounded by H we have $|H^+ \cap P_i| = \alpha_i$.*

Proof of Theorem 7. We lift all the points in C to the moment curve. The points are in general position [15] (and thus also in weak general position). By Lemma 5 if C is n-separable, then the point sets lifted to the moment curve are well-separated.

By the α-Ham-Sandwich theorem there exists a unique $(\lceil \frac{|c_1|}{2} \rceil, \ldots, \lceil \frac{|c_n|}{2} \rceil)$-cut that halves all colors. This cut is a hyperplane H that goes through n lifted points, one point of each color. These points define a solution $Q = (q_1, \ldots, q_n)$ of 2-Thief-Necklace-Splitting.

Assume that the solution Q is not unique, i.e., there is another solution $Q' \neq Q$ to C. The points Q' lifted to the moment curve define another hyperplane $H' \neq H$ with one point of each color, which is also a $(\lceil \frac{|c_1|}{2} \rceil, \ldots, \lceil \frac{|c_n|}{2} \rceil)$-cut. But by Lemma 8 there is a unique hyperplane with this property, so Q' cannot exist. ◀

[1] Weak general position is a condition that requires only subsets of the points of the form $\{p_1, \ldots, p_n, p_{n+1}\}$ for $p_i \in P_i$ for $1 \leq i \leq n$ and $p_{n+1} \in P_1 \cup \ldots \cup P_n$ to be in general position.

In this proof, we do not use the property that Lemma 8 guarantees that there is a solution for *every* choice of α, we merely use it for the guaranteed uniqueness of a solution for a halving cut.

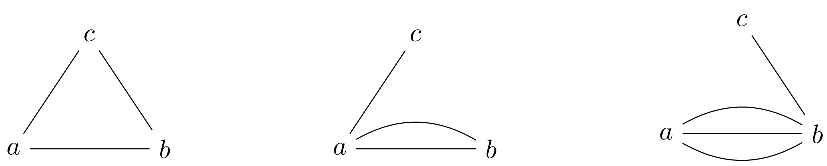
Note that the opposite direction of Theorem 7 does not hold, i.e., there are necklaces with n colors which are not n-separable but still have unique solutions for 2-Thief-Necklace-Splitting, see Figure 2c for an example.

2.2 Graph-Theoretic Aspects

To argue about the separability of necklaces, we wish to think about graphs rather than strings or even point sets. For every necklace, we thus define its walk graph:

▶ **Definition 9** (Walk graph). *Given a necklace C, the walk graph G_C is the multigraph with $V = C$ and with every potential edge $\{a, b\} \in \binom{V}{2}$ having multiplicity equal to the number of substrings "ab" plus the number of substrings "ba" in the string describing C.*

The walk graphs of the example necklaces in Figure 1 can be seen in Figure 3.



(a) Walk graph for "a b c a". **(b)** Walk graph for "a b a c". **(c)** Walk graph for "a b a b c".

Figure 3 Walk graphs of the examples in Figure 1.

Note that given a necklace C as a set of point sets, both the string describing it as well as the walk graph can be built in linear time in the size of the necklace $\sum_{c \in C} |c|$.

Recall that a graph is *Eulerian* if it contains a Eulerian tour, a closed walk that uses all edges exactly once. A graph is *semi-Eulerian* if it contains a Eulerian path, a (not necessarily closed) walk that uses all edges exactly once.

▶ **Observation 10.** *The walk graph of a necklace is connected and semi-Eulerian.*

Recall the following well-known fact about semi-Eulerian graphs.

▶ **Lemma 11.** *In a semi-Eulerian (multi-)graph, at most two vertices have odd degrees.*

The separability of a necklace turns out to be equivalent to the max-cut in its walk graph.

▶ **Definition 12** (Cut). *In a (multi-)graph G on the vertices V, a* cut *is a subset $A \subseteq V$. The* size $\mu(A)$ of a cut *A is the number of edges $\{u, v\}$ in G such that $u \in A$ and $v \notin A$. The* max-cut*, denoted by $\mu(G)$, is the largest size of any cut $A \subseteq V$.*

▶ **Lemma 13.** *For every necklace C, we have $sep(C) = \mu(G_C)$.*

Proof. For every subset $A \subseteq C$, the number of separator points needed to separate the colors in A from $C \setminus A$ is given by the size of the cut A in G_C, since the edges going over this cut correspond one to one to the points in the necklace where the necklace switches from a color in A to a color not in A, or vice versa. Thus, the max-cut $\mu(G_C)$ corresponds to the maximal number of separator points we need to separate any two subsets of colors. ◀

In our proofs we will often show that certain structures or properties do not appear in walk graphs of necklaces with bounded separability. The general strategy for these proofs will be to show that walk graphs with these structures or properties have a large max-cut, and thus the corresponding necklaces cannot have the claimed separability. Our main tool for this is the following bound, originally conjectured by Erdős [10] and proven by Edwards [8, 9].

▶ **Theorem 14** (Edwards-Erdős bound). *A simple connected graph G with n vertices and m edges has a maximum cut $\mu(G)$ of at least $\omega(G) := \frac{m}{2} + \frac{n-1}{4}$.*

Since walk graphs are not simple graphs, we will use a corollary of the following strengthening, due to Poljak and Turzík [16]:

▶ **Theorem 15** ([16]). *For a connected graph G with weight function $w : E \to \mathbb{R}_+$, there exists a cut of weight at least*

$$\frac{\sum_{e \in E} w(e)}{2} + \frac{t(G,w)}{4},$$

where $t(G,w)$ is the weight of a minimum-weight spanning tree of G.

▶ **Corollary 16.** *A connected (multi-)graph G with n vertices and m edges has a maximum cut $\mu(G)$ of at least $\omega(G) := \frac{m}{2} + \frac{n-1}{4}$.*

For determining the separability of a necklace, we will use an algorithm due to Crowston, Jones and Mnich [6] to decide max-cut beyond the Edwards-Erdős bound.

▶ **Theorem 17** (FPT algorithm [6]). *There exists an algorithm that decides whether for a given simple connected graph G with n vertices and m edges the max-cut $\mu(G)$ is at most $\omega(G) + k$ in time $2^{O(k)} \cdot n^4$.*

This is a so-called fixed-parameter algorithm; for any *fixed* parameter k, the algorithm runs in polynomial time in n, with the exponent not depending on k. Note again that this algorithm only works on simple graphs, thus, we will need to alter the walk graphs to be able to apply this algorithm.

3 An FPT Algorithm for 2-Thief-Necklace-Splitting

In this section we show Theorem 1:

▶ **Theorem 1.** *2-Thief-Necklace-Splitting can be solved in time $2^{O(\ell \log \ell)} + O(m^2)$ on every $(n-1+\ell)$-separable necklace C with n types of jewels and m total jewels.*

The algorithm we use is recursive, based on the following crucial observation.

▶ **Theorem 18.** *Let C be an $(n-1+\ell)$-separable necklace with n colors. If $n \geq 6\ell + 2$ there must exist*

(i) *two neighboring colors that are both intervals, or*
(ii) *one color that only consists of exactly two components.*

Proof. Since the walk graph is semi-Eulerian, it contains either 0 or 2 vertices with odd degree (recall Observation 10 and Lemma 11). A color that is an interval has degree 2, unless it is at the beginning or end of the necklace. A color that consists of more than two components has degree at least 6 (or 5 or 4 if it is at the beginning and/or end of the necklace).

Let $A \subseteq C$ be the set of intervals. Note that if no two intervals are neighboring, we can pick all the intervals as a cut A, which has size at least $\mu(A) \geq 2|A| - 2$. Since we know that $\mu(G_C) \leq n - 1 + \ell$, we must have that $|A| \leq \frac{n+1+\ell}{2}$.

Assume that the theorem does not hold, and that there thus exist no neighboring intervals and no color consisting of exactly two components. We can then bound the sum of degrees $\sum_{c \in C} deg(c) \geq 2 \cdot \frac{n+1+\ell}{2} + 6 \cdot (n - \frac{n+1+\ell}{2}) - 2 = 4n - 2\ell - 4$. Thus, the number of edges $|E|$ in G_C is bounded $|E| \geq \frac{4n-2\ell-4}{2} = 2n - \ell - 2$. Due to Corollary 16 we thus get that $\mu(G_C) \geq \frac{2n-\ell-2}{2} + \frac{n-1}{4} = \frac{5}{4}n - \frac{\ell}{2} - \frac{5}{4}$. By the assumption $n \geq 6\ell + 2$, we therefore have $\mu(G_C) \geq n - \frac{3}{4} + \ell$, which is a contradiction to the assumption that $\mu(G_C) \leq n - 1 + \ell$. Thus, the theorem follows. ◀

To use Theorem 18 to recursively solve smaller instances, we need to make sure that the separability of the smaller instances translates back to the separability of the original instance. The following two lemmas provide this necessary correspondence.

▶ **Lemma 19.** *Let C be a necklace. Let C' be the necklace obtained by removing two neighboring intervals c, c' from C. Then, $sep(C') = sep(C) - 2$.*

Proof. In the walk graph, removing two neighboring intervals corresponds to replacing a path (a, c, c', b) of length 3 by a direct edge connecting a and b.

Every cut $A' \subseteq C'$ in $G_{C'}$ of size k can be extended to a cut $A \subseteq C$ in G_C of size $k+2$: For every vertex $v \in C'$, we have $v \in A'$ iff $v \in A$. Furthermore, $c \in A$ iff $a \notin A'$ and $c' \in A$ iff $b \notin A'$. Thus, $sep(C) \geq sep(C') + 2$.

Similarly, every cut $A \subseteq C$ in G_C of size k induces a cut $A' = A \cap C'$ of size at least $k - 2$ in $G_{C'}$. Thus, $sep(C') \geq sep(C) - 2$, and we get $sep(C') = sep(C) - 2$. ◀

▶ **Lemma 20.** *Let C be a necklace on n colors that is $(n-1+\ell)$-separable. The necklace C' obtained by reducing a color $c \in C$ to a subset $\emptyset \subset c' \subset c$ is still $(n-1+\ell)$-separable.*

Proof. By simplifying a necklace, we cannot increase its separability. ◀

We are now ready to present Algorithm 1, an FPT algorithm to solve 2-Thief-Necklace-Splitting on $(n-1+\ell)$-separable necklaces. The strategy is to reduce the given necklace either by removing two neighboring intervals, or by removing one of the two components in a color that consists of exactly two components. By Lemmas 19 and 20, if C is $(n-1+\ell)$-separable, the resulting necklace C' is again $(n'-1+\ell)$-separable (for $n' = |C'|$), and can thus be solved recursively. The solution of the reduced case is then extended back to a solution of the original necklace. A necklace can be reduced as long as Theorem 18 applies, and thus we only need to solve the case $n < 6\ell + 2$ directly.

For an example of the execution of the algorithm, see Figure 4 and Figure 5. Note that these small instances would technically be solved by brute-force and merely serve as illustrations.

Proof of Theorem 1. We first argue for correctness of Algorithm 1. By Theorem 18, if we reach line 8 we can always find a color which consists out of exactly two components, so the algorithm can never fail to finish.

We have to argue that our algorithm returns a correct solution in both line 6 and line 15.

(i) Line 6: The constructed solution splits the two neighboring intervals correctly. Since we place two splits, the parity of the partition outside of these intervals does not change in comparison to the solution Q obtained recursively. Thus, all other colors are also split correctly.

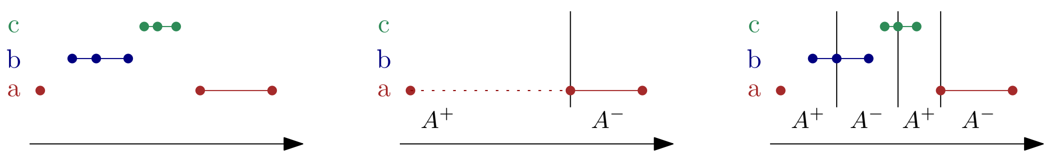
Algorithm 1 RecursiveNS.

Input: An $(n-1+\ell)$-separable necklace C with n colors.
Output: n split points.
1: **if** $n < 6\ell + 2$ **then**
2: $Q \leftarrow$ BruteForce(C)
3: **return** Q
4: **else if** there exist two neighboring intervals $c, c' \in C$ **then**
5: $Q \leftarrow$ RecursiveNS$(C \setminus \{c, c'\})$
6: **return** $Q \cup \{median(c), median(c')\}$
7: **else**
8: $c \leftarrow$ a color consisting of two components c_1, c_2
9: $c' \leftarrow$ largest component of c
10: **if** $|c'|$ is even **then**
11: Add a median point to c'
12: $Q \leftarrow$ RecursiveNS$((C \setminus \{c\}) \cup \{c'\})$
13: $\{q\} \leftarrow Q \cap c'$
14: $q' \leftarrow q$ shifted right/left by $\lceil \frac{min(|c_1|,|c_2|)}{2} \rceil$ points of c' ▷ *direction depending on parity of number of split points in Q between c_1 and c_2*
15: **return** $Q \setminus \{q\} \cup \{q'\}$

(ii) Line 15: The constructed solution splits color c correctly, and q' lies in the same component of c as q, since c' is the larger of the two components. Shifting the split within the same component of c does not change the partition outside of this component in comparison to the solution Q obtained recursively. Thus, all other colors are also split correctly.

It remains to argue for the runtime of Algorithm 1. Clearly, we only use the brute-force approach at line 2 once. In an $(n-1+\ell)$-separable necklace with $n < 6\ell + 2$, each color has at most $O(\ell)$ components. For each guess of one component per color, it can be determined in polynomial time in ℓ whether this guess admits a solution. There are at most $\ell^{O(\ell)}$ guesses, thus we can solve this base case in time $2^{O(\ell \log \ell)}$.

In the rest of the algorithm, on each level of the recursion we reduce the number of points in the necklace by at least one, and we can make the necessary adjustments and find the needed colors in linear time in the number of points. Thus, the total runtime of the algorithm is $2^{O(\ell \log \ell)} + O(\sum_{c \in C} |c|)^2$, as claimed. ◀

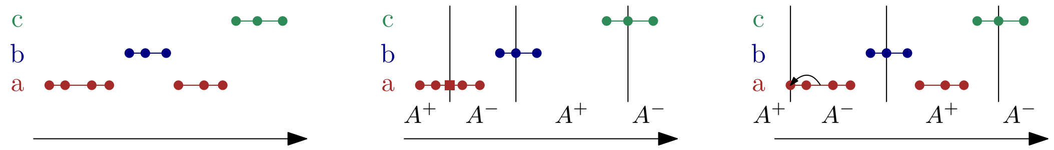


(a) Original necklace. **(b)** Reduced necklace. **(c)** Solution.

Figure 4 Example step of Algorithm 1 using the reduction of removing two neighboring intervals (b and c).

For the special case of n-separable necklaces, i.e., $\ell = 1$, we get the following corollary:

▶ **Corollary 21.** *Finding the unique solution for 2-Thief-Necklace-Splitting on an n-separable necklace with n colors takes polynomial time.*



(a) Original necklace. **(b)** Reduced necklace. **(c)** Solution.

Figure 5 Example step of Algorithm 1 using the reduction of removing a component from the two-component color a.

Until now, both Theorem 1 and Corollary 21 work under the initial promise that C is $(n-1+\ell)$-separable (or n-separable respectively). If the algorithm fails because none of the cases applies, this certifies that the input necklace was *not* $(n-1+\ell)$-separable. On the other hand, Algorithm 1 may run successfully, even if the input necklace is not $(n-1+\ell)$-separable, and if it does run successfully, its output is always a correct solution. Since Algorithm 1 can produce these "false positives", it cannot be used to decide $(n-1+\ell)$-separability. We tackle that problem in the next section.

4 Testing Separability

At first, it seems like finding a polynomial-time algorithm for deciding whether a necklace is $(n-1+\ell)$-separable may be futile, since we have the following theorem due to Guruswami [13]:

▶ **Theorem 22** ([13]). *Given a Eulerian graph G and an integer k, deciding whether the size of the max-cut $\mu(G) \geq k$ is* NP*-complete.*

Since to compute the separability of a necklace we need to compute the max-cut of its walk graph, and since every Eulerian graph is the walk graph of some necklace[2], we get the following corollary:

▶ **Corollary 23.** *Given a necklace C of n colors and an integer k, deciding whether C is k-separable is* co-NP*-complete.*

However, not all hope is lost. To check whether a necklace is $(n-1+\ell)$-separable, we do not need to compute the max-cut of its walk graph, we merely need to check whether it is at most $(n-1+\ell)$. We next provide an FPT algorithm that checks $(n-1+\ell)$-separability for fixed parameter ℓ. With $\ell = 1$ this shows that testing n-separability of n colors is solvable in polynomial time, even though both testing k-separability of n colors with k as input as well as testing well-separation of point sets are co-NP-complete [4]. More generally, we show the following theorem:

▶ **Theorem 24.** *There exists an FPT algorithm for fixed parameter ℓ that can decide whether the max-cut of a given semi-Eulerian multigraph G_C with n vertices is at most $n-1+\ell$, i.e., it can decide whether $\mu(G_C) \leq n-1+\ell$ in time $2^{O(\ell^2)} \cdot n^4$.*

By Theorem 17, there exists an algorithm that decides whether a *simple* graph G with n vertices and a fixed parameter k has a max-cut of size $\mu(G) \leq \omega(G) + k = \frac{|E(G)|}{2} + \frac{n-1}{4} + k$ in $2^{O(k)} \cdot n^4$ time. But our input graph G_C is a multigraph and we have no bound on its number of edges, nor on the distance between $\omega(G_C)$ and $n-1+\ell$. In order to use this algorithm to decide separability, we need the following:

[2] Simply find a Eulerian path through the graph and place one point per character in the respective color. If some color has an even number of points, add one more to an existing component.

1. Derive a graph G'_C from G_C such that we *can* lower bound $|E(G'_C)|$ and thus $\omega(G'_C)$.
2. Prove that there is a bounded number of multi-edges in G'_C.
3. Transform G'_C into a simple graph G''_C by blowing up its multi-edges by a constant factor.

In the following, we will use the term *interval* for intervals on necklaces as well as their corresponding vertices interchangeably. The intervals in C correspond to the vertices in G_C with degree at most 2, except for possibly one vertex of degree 2 that is both the starting and ending point of the fixed Eulerian path; this single vertex does not correspond to an interval.

▶ **Lemma 25.** *Given a semi-Eulerian multigraph G on n vertices, we can either detect that $\mu(G) > n-1+\ell$, or we can build a multigraph G' on n' vertices such that $|E(G')| \geq \frac{3}{2}n' - \frac{\ell}{2} - 1$, and such that $\mu(G) \leq n - 1 + \ell$ if and only if $\mu(G') \leq n' - 1 + \ell$.*

Proof. Given a multigraph G, let G' be the result of applying Lemma 19 on G exhaustively. As long as there are two adjacent intervals in G, we can remove the two intervals, thus reducing the maximum cut size by 2. In each such step we remove 2 vertices, 3 edges and add 1 new edge. Thanks to Lemma 19, we have the desired correspondence between $\mu(G)$ and $\mu(G')$.

Assume there are at least $\frac{n'+\ell}{2} + 1$ intervals in G'. Then the cut A in G' with all intervals on one side and all other vertices on the other side has size $\mu(A) \geq 2 \cdot (\frac{n'+\ell}{2} + 1) - 2 = n' + \ell$. It follows that $\mu(G') \geq \mu(A) > n' - 1 + \ell$. In this case we can thus detect that $\mu(G) > n - 1 + \ell$.

In the other case, there are strictly fewer than $\frac{n'+\ell}{2} + 1$ intervals in G'. All other vertices have degree at least 4 (excluding the start and end vertex). Therefore the sum of degrees in G' is

$$\sum_{v \in V(G')} deg(v) \geq \frac{n' + \ell}{2} \cdot 2 + \frac{n' - \ell}{2} \cdot 4 - 2 = 3n' - \ell - 2.$$

Thus the number of edges in G' is $|E(G')| \geq \frac{3}{2}n' - \frac{\ell}{2} - 1$. ◀

We can now see the following.

▶ **Observation 26.** *Given this bound on $|E(G')|$, the bound $\omega(G')$ given by Corollary 16 can be bounded by*

$$\omega(G') \geq \frac{\frac{3}{2}n' - \frac{\ell}{2} - 1}{2} + \frac{n' - 1}{4} = n' - \frac{\ell}{4} - \frac{3}{4}.$$

Thus, by the process of eliminating neighboring intervals, we have managed to get the difference between $(n' - 1 + \ell)$ and $\omega(G'_C)$ to be a constant depending only on ℓ.

Next we show that the total multiplicity M of the multi-edges in G'_C cannot be too large. We show that if G'_C has maximum cut size at most $n' - 1 + \ell$, the total multiplicity of multi-edges can be bounded by a function solely depending on ℓ, and not n or $|E(G'_C)|$.

▶ **Lemma 27.** *In a multigraph G on n vertices with $\mu(G) \leq n - 1 + \ell$, the total multiplicity of the multi-edges in G is at most $2\ell^2$.*

Proof. Let G' be a weighted simple graph with an edge of weight $m - 1$ for every multi-edge of multiplicity $m \geq 2$ in the graph G. Note that the total weight of G' is at least half of the total multiplicity of multi-edges in G.

Let F be a spanning forest in G' with total weight w. Given F, we can build a spanning tree T of G of total weight $n - 1 + w$, since every edge of F of weight $m' - 1$ corresponds to a multi-edge of multiplicity m' in G, and all additional edges used to make F into a spanning tree have weight 1. Since every tree is bipartite, the weight of T is a lower bound on the max-cut of G: $\mu(G) \geq n - 1 + w$. Thus, for a given G with $\mu(G) \leq n - 1 + \ell$, the total weight of F must be at most ℓ.

We thus only need to show that in a simple weighted graph (in our case, G'), in which every weight is at least 1, and whose maximum-weight spanning forest has weight at most ℓ, the total weight of the graph is at most ℓ^2. To see this, we successively remove spanning forests from G' until G' is empty. Every spanning forest we remove has weight at most ℓ. As every edge has weight at least 1, every vertex in G' has degree at most ℓ. Thus, we are done after removing at most ℓ spanning forests. Thus, the total weight of G' is at most ℓ^2.

We conclude that the total multiplicity of multi-edges in G can be at most $2\ell^2$. ◀

Finally, we show how G'_C can be transformed into a simple graph G''_C. Let a and b be vertices in G'_C with a multi-edge of multiplicity m between them. We construct the graph G''_C from G'_C by removing the multi-edge between a and b and introducing m paths of length three from a to b, all going through separate vertices. See Figure 6 for an example application of this process.

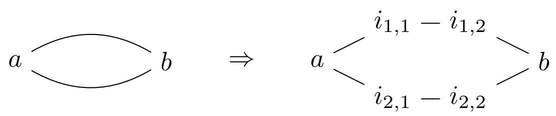


Figure 6 Example of blowing up a multi-edge of multiplicity 2 to make the graph simple.

This process is again constructed in such a way that the change of the max-cut is predictable:

▶ **Lemma 28.** *Let G be a multigraph on n vertices. Let a and b be vertices in G with a multi-edge of multiplicity m between them. Let G' be the result of blowing up the multi-edge between a and b in G. Then, $\mu(G') = \mu(G) + 2m$.*

Proof. Let $A \subseteq V(G)$ be some max-cut in G with $\mu(A) = \mu(G)$.

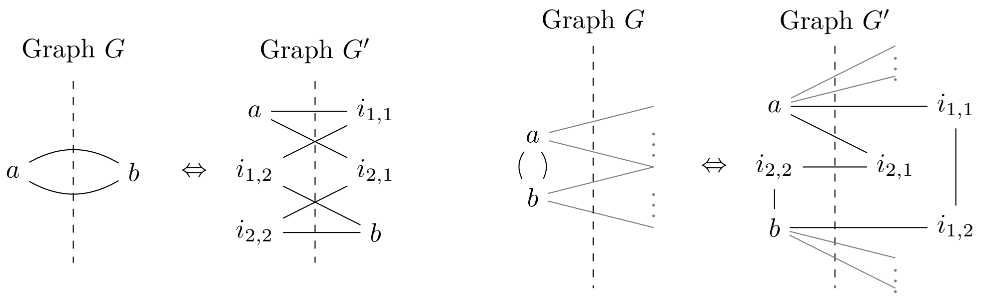
We distinguish between two cases. If the multi-edge goes across the cut, i.e. $a \in A$ and $b \notin A$, the same cut in G' has m fewer edges (namely the multi-edge) and $3m$ edges more, namely all of the newly introduced edges of the paths, see Figure 7a. If the multi-edge between a and b is *not* in the max-cut of G, there is a cut in G' that has $2m$ new edges, namely one of each newly introduced path, see Figure 7b. Thus, a max-cut of size $\mu(G)$ in G implies a cut of size $\mu(G) + 2m$ in G', and thus $\mu(G') \geq \mu(G) + 2m$.

For the other direction, consider a max-cut A' of G'. Since A' is maximal, it must either contain all $3m$ intermediate edges between a and b, and put a and b on different sides of the cut, or it must put a and b on the same side of the cut, and contain exactly $2m$ intermediate edges (see again Figure 7). Thus, there must exist a cut A in G which contains exactly $2m$ fewer edges than A', and we get $\mu(G) \geq \mu(G') - 2m$.

We conclude that $\mu(G') = \mu(G) + 2m$. ◀

We are now ready to put this all together and describe the algorithm proving Theorem 24.

Proof of Theorem 24. We prove that Algorithm 2 is correct and runs in time $2^{f(\ell)} \cdot n^4$. Correctness follows from Lemma 25, Lemma 27 and Lemma 28. Clearly, all steps except the invocation of the FPT algorithm of Theorem 17 in the last line can be performed in $O(n^2 + \ell^2)$.



(a) Case 1: The multi-edge is in max-cut of G. **(b)** Case 2: The multi-edge is *not* in max-cut of G.

Figure 7 Change of max-cut size when blowing up a multi-edge of multiplicity 2.

Algorithm 2 FPT algorithm for testing $\mu(G_C) \leq n - 1 + \ell$ with fixed parameter ℓ.

Input: A semi-Eulerian multigraph G_C on n vertices.
Output: True iff $\mu(G_C) \leq n - 1 + \ell$.

$G'_C \leftarrow G_C$
while there exist neighboring intervals in G'_C **do**
  Remove two neighboring intervals from G'_C.
$n' \leftarrow |V(G'_C)|$
$i \leftarrow$ The number of intervals in G'_C.
if $i > \frac{n'+\ell}{2}$ **then return** false ▷ *based on Lemma 25*
$M \leftarrow$ The total multiplicity of multi-edges in G'_C.
if $M > 2\ell^2$ **then return** false ▷ *based on Lemma 27*
$G''_C \leftarrow$ The result of applying Lemma 28 to every multi-edge in G'.
return $\mu(G''_C) \leq (n' - 1 + \ell) + 2M$ ▷ *using FPT algorithm of Theorem 17.*

We choose k such that when we call the FPT algorithm of Theorem 17 with G''_C and k it decides $\mu(G''_C) \leq (n' - 1 + \ell) + 2M$, i.e., we choose k such that $(n' - 1 + \ell) + 2M = \omega(G''_C) + k$. Therefore let $k := ((n' - 1 + \ell) + 2M) - \omega(G''_C)$.

For bounding the runtime of this invocation, we need to check that k is dependent only on ℓ. Recall that by Observation 26 we can bound $(n' - 1 + \ell) - \omega(G'_C) \leq \frac{5}{4}\ell - \frac{1}{4}$, a quantity depending only on our parameter ℓ. To relate $\omega(G''_C)$ to $\omega(G'_C)$, we can see that blowing up a multi-edge in G' of multiplicity m adds $2m$ edges and $2m$ vertices and thus changes ω by $\frac{2m}{2} + \frac{2m}{4} = \frac{3}{2}m$. Thus we have $\omega(G''_C) = \omega(G'_C) + \frac{3}{2}M$. We can now put everything together and get $k \leq 2M + \frac{5}{4}\ell - \frac{1}{4} - \frac{3}{2}M = \frac{1}{2}M + \frac{5}{4}\ell - \frac{1}{4}$, and since $M \leq \ell^2$, we get that k is bounded by $O(\ell^2)$. Thus, the final invocation of the algorithm of Theorem 17 runs in time $2^{O(\ell^2)} \cdot n^4$. ◀

5 Conclusion and Further Directions

In conclusion, we proved that 2-Thief-Necklace-Splitting on n-separable necklaces has a unique solution and can be solved in polynomial time. Also n-separability can be tested in polynomial time. Furthermore, we showed that 2-Thief-Necklace-Splitting, which in general is known to be PPA-complete, admits an FPT algorithm for the parameter ℓ such that the input necklace is $(n - 1 + \ell)$-separable. Lastly, we showed that testing $(n - 1 + \ell)$-separability is also FPT, even though testing well-separation of point sets in $\mathbb{R}^n$ is co-NP-complete.

The condition of n-separability is only sufficient for uniqueness of the solution to 2-Thief-Necklace-Splitting. An interesting followup question is whether there also exist necessary conditions for such uniqueness.

As our main open question we wonder how our algorithm for 2-Thief-Necklace-Splitting can be extended to more general settings. Firstly, can we also find polynomial time algorithms for k-Thief-Necklace-Splitting under the constraint of n-separability? Secondly, instead of halving every color class, can we maybe find an algorithm to find any $(\alpha_1, \dots, \alpha_n)$-cut? The existence of these cuts is also guaranteed by Lemma 8, however our algorithm really only works for halving, since if we are not halving, the solution is not guaranteed to split a color with two components in the bigger component.

Another interesting followup question is whether one can lift the definition of k-separability into higher dimensions. In other words, for n point sets $P = \{P_1, \dots, P_n\}$ in $\mathbb{R}^d$, can each subset A of P be separated from $P \setminus A$ by k hyperplanes? Well-separation then becomes 1-separability. Thus, deciding k-separability for k as input or even for the case $k = 1$ is co-NP-hard. It is likely that special cases such as d-separability or n-separability are also hard to decide. While well-separation is also contained in co-NP, this is not clear for k-separability for $k > 1$. Like 2-Thief-Necklace-Splitting, which has a unique solution under the condition of n-separability, one could also investigate whether there are other geometric problems which gain interesting properties under the condition of the input being well-separated, or k-separable for some k.

Finally, can we extend our FPT algorithm for deciding $\mu(G) \leq n - 1 + \ell$ on semi-Eulerian multigraphs to work on all connected multigraphs? Furthermore, can we maybe also decide $\mu(G) \leq \omega(G) + \ell$ (to get a direct analogue of the algorithm of Crowston, Jones, and Mnich for multigraphs) and not just $\mu(G) \leq n - 1 + \ell$?

References

1 Noga Alon. Splitting necklaces. *Advances in Mathematics*, 63(3):247–253, 1987. doi:10.1016/0001-8708(87)90055-7.

2 Noga Alon and Andrei Graur. Efficient splitting of necklaces. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.14.

3 Noga Alon and Douglas B. West. The Borsuk-Ulam theorem and bisection of necklaces. *Proceedings of the American Mathematical Society*, 98(4):623–628, 1986. doi:10.1090/S0002-9939-1986-0861764-9.

4 Helena Bergold, Daniel Bertschinger, Nicolas Grelier, Wolfgang Mulzer, and Patrick Schnider. Well-Separation and Hyperplane Transversals in High Dimensions. In Artur Czumaj and Qin Xin, editors, *18th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2022)*, volume 227 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:14, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SWAT.2022.16.

5 Man-Kwun Chiu, Aruni Choudhary, and Wolfgang Mulzer. Computational Complexity of the α-Ham-Sandwich Problem. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:18, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2020.31.

6 Robert Crowston, Mark Jones, and Matthias Mnich. Max-cut parameterized above the Edwards-Erdős bound. *Algorithmica*, 72(3):734–757, 2015. doi:10.1007/s00453-014-9870-z.

7 Jesús De Loera, Xavier Goaoc, Frédéric Meunier, and Nabil Mustafa. The discrete yet ubiquitous theorems of Carathéodory, Helly, Sperner, Tucker, and Tverberg. *Bulletin of the American Mathematical Society*, 56(3):415–511, 2019. doi:10.1090/bull/1653.

8 Christopher S. Edwards. Some extremal properties of bipartite subgraphs. *Canadian Journal of Mathematics*, 25(3):475–485, 1973. doi:10.4153/CJM-1973-048-x.

9 Christopher S. Edwards. An improved lower bound for the number of edges in a largest bipartite subgraph. In *Proc. 2nd Czechoslovak Symposium on Graph Theory, Prague*, pages 167–181, 1975.

10 Paul Erdős. On some extremal problems in graph theory. *Israel Journal of Mathematics*, 3:113–116, 1965.

11 Aris Filos-Ratsikas and Paul W. Goldberg. The complexity of splitting necklaces and bisecting ham sandwiches. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 638–649, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3313276.3316334.

12 Charles H. Goldberg and Douglas B. West. Bisection of circle colorings. *SIAM Journal on Algebraic Discrete Methods*, 6(1):93–106, 1985. doi:10.1137/0606010.

13 Venkatesan Guruswami. Maximum cut on line and total graphs. *Discrete Applied Mathematics*, 92(2):217–221, 1999. doi:10.1016/S0166-218X(99)00056-6.

14 Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Separations in proof complexity and TFNP. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1150–1161, Los Alamitos, CA, USA, 2022. IEEE Computer Society. doi:10.1109/FOCS54457.2022.00111.

15 Jiří Matoušek. *Lectures on Discrete Geometry*. Graduate Texts in Mathematics. Springer New York, 2002. doi:10.1007/978-1-4613-0039-7.

16 Svatopluk Poljak and Daniel Turzík. A polynomial time heuristic for certain subgraph optimization problems with guaranteed worst case bound. *Discrete Mathematics*, 58(1):99–104, 1986. doi:10.1016/0012-365X(86)90192-5.

17 Sambuddha Roy and William Steiger. Some combinatorial and algorithmic applications of the Borsuk–Ulam theorem. *Graphs and Combinatorics*, 23(1):331–341, June 2007. doi:10.1007/s00373-007-0716-1.

18 William Steiger and Jihui Zhao. Generalized ham-sandwich cuts. *Discrete & Computational Geometry*, 44(3):535–545, 2010. doi:10.1007/s00454-009-9225-8.

19 Arthur H. Stone and John W. Tukey. Generalized "sandwich" theorems. *Duke Math. J.*, 9(2):356–359, 1942. doi:10.1215/S0012-7094-42-00925-6.