Weighted Genome Rearrangements

Pijus Simonaitis

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITE DE MONTPELLIER

En Informatique

École doctorale : Information, Structures, Systèmes

Unité de recherche LIRMM UMR 5506

Weighted Genome Rearrangements

Présentée par Simonaitis Pijus
Le 10/07/2020

Sous la direction de Annie CHATEAU et Krister SWENSON

Devant le jury composé de

Berry Vincent, Professeur des universités, Université de Montpellier	Examinateur
Chateau Annie, Maîtresse de Conférence, Université de Montpellier	Directrice de Thèse
Fertin Guillaume, Professeur des universités, Université de Nantes	Rapporteur
Meidanis Joao, Professor, University of Campinas	Examinateur
Middendorf Martin, Professor, Leipzig University	Rapporteur
Swenson Krister, Chargé de recherche, Université de Montpellier	Co-encadrant de Thèse

UNIVERSITÉ DE MONTPELLIER

Abstract

Recent advances in sequencing technologies revealed the ubiquity of genome rearrangements between each and every one of us. These large-scale mutations rearrange segments of chromosomes and have a profound impact on genetic variation, disease, and evolution. The study of the consequences of rearrangements along with their molecular mechanisms, however, is still in its infancy.

Given extant genomes, we are interested in tracing back the evolutionary rearrangement scenarios that transformed their least common ancestor into the genomes that we observe today. This not only helps to reveal evolutionary relationships between organisms, but also provides a window for the study of genome rearrangements themselves.

The central computational problem in this subfield of comparative genomics is that of finding optimal rearrangement scenarios transforming one genome into another. Historically all rearrangements were treated as being equally possible, and optimal scenarios were those that contained the minimum number of rearrangements. Recent advances in biology, however, allow us to devise much more sophisticated models. We present a short survey of the existing work on using biological constraints for genome rearrangements, and argue that a much more flexible approach is necessary to accompany the influx of newly available biological data.

In this work we propose an extremely general framework for genome rearrangements with biological constraints. Our main contribution is a polynomial time algorithm that, for an arbitrary cost function, finds a minimum cost scenario among those of minimum length. Along the way we establish a number of novel links between sorting genomes with double cut and join rearrangements, sorting graphs with 2-breaks or edge swaps, sorting permutations with mathematical transpositions, sorting strings with interchanges, and token swapping on graphs.

Keywords: weighted genome rearrangements, double cut and join, edge swap, minimum length transposition decomposition, minimum weight quadrangulation.

Résumé

Un réarrangement génomique est une mutation qui modifie la structure des chromosomes voire même leur nombre dans un génome. Outre des fusions et des fissions de chromosomes, ces réarrangements comprennent des délétions, des insertions et des inversions de segments chromosomiques. Deux extrémités de chromosomes différents peuvent également être échangées au cours d'une translocation. L'ensemble de ces mutations constitue un scénario évolutif de réarrangements entre les espèces. Nous nous sommes intéressés à la reconstruction des scénarios de réarrangements entre espèces animales.

Notre projet associe des outils mathématiques et algorithmiques avec la compréhension biologique actuelle des réarrangements génomiques. D'un point de vue biologique, notre objectif est de lier génétique et épigénétique aux réarrangements dans les deux sens :

- nous développons une méthodologie pour étudier des caractéristiques génétiques et épigénétiques associées aux réarrangements,
- et inversement pour trouver des scénarios de réarrangements guidés par de telles caractéristiques génétiques et épigénétiques.

La principale contribution de cette thèse est la suivante. Nous présentons un cadre sur le modèle de réarrangements double cut and join avec des poids arbitraires. Dans ce cadre un scénario de poids minimum peut être trouvé en temps polynomial parmi les scénarios de longueur minimale pour deux génomes à contenu génétique identique et sans doublons.

En plus de cela, nous établissons un certain nombre de nouvelles correspondances entre les divers problèmes de tri. Ces problèmes incluent le tri des génomes avec des réarrangements dits double cut and join, le tri des graphes avec 2-breaks ou edge swaps, le tri des permutations avec des transpositions, le tri des chaînes avec des échanges et l'échange de jetons sur les graphes.

Mots clés: scénario évolutif de réarrangements, double cut and join, réarrangements génomiques pondérés, décomposition de la permutation en transpositions, quadrangulation de poids minimum.

Contents

Symbols

u, v, w, s, r	a vertex
col	a color (black or gray)
$(\{u, v\}, col)$	a colored edge
G, H	a 2-edge-colored Eulerian multigraph
$\overline{G}$	a terminal graph
τ	a 2-break
ρ	a 2-break scenario or an $\mathcal{O}$-scenario
$d_{2b}(G)$	the minimum length of a 2-break scenario for a graph
$d_{b2b}(G)$	the minimum length of a black-2-break scenario for a graph
$\mathcal{H}$	an Eulerian decomposition or a labeled Eulerian decomposition
$c(G)$	the size of a Maximum Alternating Edge-disjoint Cycle Decomposition of a graph
$e(G)$	the number of edges in a graph divided by two
A, B, C	a genome
a, b, c, d	a gene extremity or a vertex label
$d_{DCJ}(A, B)$	the minimum length of a DCJ scenario for genomes
$AG(A, B)$	the adjacency graph of genomes
$G(A, B)$	the genome breakpoint graph
$\circ$	the vertex of a breakpoint graph of black and gray degrees higher than one
n	the number of genes in a genome or the number of vertices in a graph
δ	a double cut and join
Δ	a double cut and join scenario or an Eulerian tour
σ	a permutation
id	the identity permutation
π	a transposition
$H(\sigma_1, \sigma_2)$	the permutation breakpoint graph
$d_{Cayley}\sigma_1, \sigma_2)$	Cayley distance between permutations
T	a transposition decomposition of a permutation

$H(A, B)$	the internal genome breakpoint graph
$\mathcal{D}(G, \rho)$	the trajectory graph of a graph and its 2-break scenario
S	a simple cycle
$\overrightarrow{G}$	an Eulerian orientation of a graph
C	a circle
$C[i, j]$	a sub-circle
$\mathcal{S}(C, \rho)$	the scenario graph of a circle and its 2-break scenario
$\mathcal{M}(C, \rho)$	the scenario matching of a circle and its 2-break scenario
Σ_C	a circular straight-line embedding of a circle
Σ_V, Σ_E	alphabets of edge and vertex labels
x, y, z, t, q	an edge label
$\mathcal{O}$	a set of valid operations
$\lambda = (\lambda_V, \lambda_E)$	a labeling of graph's vertices and edges
$(\{u, v\}, col, x)$	a labeled edge
$d_{\mathcal{O}b}(G, \lambda)$	the minimum 2-break-length of an $\mathcal{O}$-scenario for a labeled graph
χ	a labeling of 2-break's edges
(τ, χ)	an $\mathcal{O}$-break
φ	a positive real valued cost function on a set of valid operations $\mathcal{O}$
$\varphi(\rho)$	φ-cost of an $\mathcal{O}$-scenario
$\mathrm{MCS}_\varphi(G, \lambda)$	the minimum φ-cost of an $\mathcal{O}$-scenario for a labeled graph
$\mathrm{MCPS}_\varphi(G, \lambda)$	the minimum φ-cost of a parsimonious $\mathcal{O}$-scenario for a labeled graph
(H, λ_H)	a labeled subgraph of a labeled graph (G, λ)
$col_{\{i,j\}}$	the color of a colored outer edge of a sub-circle $C[i, j]$
$(C[i, j], \lambda^x)$	a labeled sub-circle with the label of the colored outer edge equal to x
L	the number of edge labels

Chapter 1

Introduction

1.1 Genome Rearrangements in an Evolutionary Setting

A *rearrangement*, also known as a *structural variant*, is a large-scale mutation that modifies the structure of the chromosomes or even their number in a genome. Take for example humans and our closest living relatives chimpanzees. We have 23 pairs of chromosomes, while chimps have 24. This difference is due to a fusion of two non-human ancestral primate chromosomes that resulted in human chromosome 2 [36, 37]. Studying this particular event can inform us about human evolution in multiple ways. Did this fusion lead to a speciation event that separated human and chimp lineages, or maybe those of archaic humans? Did it have a functional impact or trigger an advent of any phenotypic changes? If we were able to accurately date this fusion, could it be used for dating other events of human evolution? How exactly did it happen, why was it not lethal, and how did it spread within a population? Human and chimp lineages have each accumulated a number of other genomic rearrangements since their separation [65]. Besides chromosome *fusion* and *fission* these rearrangements include *deletions*, *insertions* and *inversions* of chromosomal segments. Two ends of different chromosomes might also get swapped during a *translocation* event. These mutations together constitute an *evolutionary scenario* between the species, and in this work we will be interested in reconstructing such scenarios between organisms.

Recent advances in sequencing technologies provide us with an unprecedented opportunity to study the mechanisms behind genome rearrangements [71, 66], the ways they spread in populations [40], and their evolutionary significance [74]. Methods for a systematic detection of genome rearrangements in populations and cancer cells are emerging [60, 92], and more and more high-quality well-annotated genome assemblies are available for comparative research. Building upon these advances

recent studies started to unveil a role played in the evolutionary rearrangement scenarios by *active chromatin*, spatial proximity of loci in a genome, *non-canonical DNA structures*, and such chromatin features as *topologically associating domains* (TADs). Rearrangements between human and mouse have been shown to occur in active chromatin regions coming into 3D proximity in the nucleus [99, 17, 94]. Gibbon rearrangements were shown to occur at TAD boundaries, with most TADs maintained as intact modules during and after a rearrangement [68]. A genome-wide depletion of deletions in active chromatin, and at TAD boundaries was observed across primate evolution [50], while deletions causing TAD fusions were shown to be rare and under negative selection in humans [61]. Transcriptional activity [53] and non-canonical DNA structures [52] were both linked to rearrangements in cancer, and the latter awaits further testing in an evolutionary setting.

Our project brings the mathematical and algorithmic tools from computer science together with the current biological understanding behind genome rearrangements. From a biological perspective, our goal is to link genetics/epigenetics to rearrangements in both directions:

- We will develop a method for studying genetic and epigenetic patterns associated with rearrangements in an evolutionary setting, and conversely
- for finding plausible evolutionary rearrangement scenarios informed by such genetic and epigenetic patterns.

1.2 Mathematical Models for Genome Rearrangements

There is a quarter century of mathematical and algorithmic work devoted to modeling rearrangements, and finding and sampling scenarios that could have transformed the gene order of one species into the gene order of another [48].

Inversions, also known as *reversals*, were first observed in fruit flies in 1921 by Sturtevant [93], and seem to be a suitable model for genome rearrangements in this species [83]. In 1995 Hannenhalli and Pevzner [54] came up with a rearrangement scenario between human and mouse genomes consisting of 131 reversals and translocations. The latter is a rearrangement that exchanges the ends of two linear chromosomes, and in an extreme case it can reproduce chromosome fusion and fission. Hannenhalli and Pevzner proved that for suitably represented genomes the minimum number of reversals [55], or reversals and translocations [54] required to transform one genome into another can be found in polynomial time. Mathematics behind their theory is quite complicated, see Bergeron [13] for a survey.

In 2005 Yancopoulos, Attie, and Friedberg [102] proposed a mathematically drastically simpler rearrangement model called *double cut and join* (DCJ). A DCJ cuts

chromosomes in one or two places, called *breakpoints*, and joins back the resulting chromosomal strands. In addition to inversions and translocations, this simple mathematical operation can reproduce such rearrangements as a circularization of a linear chromosome, and an excision of a circular chromosome out of a linear one. It is unclear if the latter types of rearrangements play any role in evolution, however circular DNA elements have been found in healthy human cells [76] and are abundant in cancer cells [98].

A *distance* is the minimum number of rearrangements required to transform one genome into another. Reversal, reversal/translocation and DCJ distances between genomes can be found in polynomial time [55, 54, 15], however most of the problems become NP-hard once more than two genomes are compared. These include the *median* problem, which is used for phylogenetic reconstruction, and asks for a genome minimizing the sum of the pairwise distances to a given set of genomes [29, 97].

Single cut or join (SCJ), an even simpler model for genome rearrangements was proposed by Feijão and Meidanis [45] in 2011. According to the authors, "this new distance measure may be of value as a speedily computable, first approximation to distances based on more realistic rearrangement models" [45]. This claim is supported with a proof that the median problem is polynomial time solvable for SCJ. New models for genome rearrangements are still being introduced. A *rank* distance for genomes modeled as matrices was introduced by Zanetti, Biller and Meidanis [105] in 2016. It relates to DCJ and is more sophisticated than SCJ while still allowing for interesting polynomial time results on the median problem as it was recently established by Chindelevitch, La and Meidanis [32].

1.3 Weighted Genome Rearrangements

Foundational models focused solely on the minimum length, or *parsimonious*, rearrangement scenarios transforming one genome into another [84, 54, 102]. However the true evolutionary scenario is likely to be non-parsimonious as discussed by Lin and Moret [69], and more recently by Biller, Guéguen, Knibbe, and Tannier [20] and Alexeev and Alekseyev [2]. In addition to this, both for reversals and DCJ the number of possible parsimonious scenarios was shown to grow exponentially with respect to the distance. These results were respectively established by Bergeron, Chauve, Hartman, and St-Onge [14], and Braga and Stoye [24]. This means that even if the true evolutionary scenario happened to be parsimonious, then one would still need some additional biological constraints on the rearrangement scenarios in order to select the ones that are more likely to resemble the true one.

To this end, the study of weighted genome rearrangements was pioneered by Blanchette, Kunisawa, and Sankoff [22] in 1996. They assigned different weights to a reversal, a *transposition* and an *inverse transposition*, where the former swaps two

contiguous segments of a chromosome while the latter also inverts one of them. A greedy algorithm for minimizing the sum of the rearrangement weights in a scenario was proposed in [22]. The rationale behind this model is that different types of rearrangements might have different chances of appearing in the true evolutionary scenario. Hartmann, Wieseke, Sharan, Middendorf, and Bernt [58] recently proposed a polynomial sized ILP for solving this problem with arbitrary weights. A variant of this problem where only transpositions and reversals are allowed, and a ratio of their weights satisfies $w_{tr}/w_{rev} \leq 1.5$ was proved to be NP-hard by Oliveira, Brito, Dias, and Dias [80].

Another approach deals with a notion of a *preserved gene cluster*, also known as a *preserved common interval*. It is a set of genes whose order might be shuffled but that remain clustered together in both genomes. This line of work is motivated by an observation that such conserved gene clusters sometimes contain functionally associated proteins [96], and thus might be unlikely to get broken by a rearrangement in the true evolutionary scenario. See Hartmann, Middendorf, and Bernt [57] for a recent survey of algorithmic work related to this constraint. The same authors in [56] provide an exact algorithm for finding a minimum weight preserving scenario, where different weights are assigned to inversions, transpositions, inverse transpositions, and *tandem duplication random loss* operations, that duplicate a segment of a chromosome and delete at random one copy of each duplicated gene. These four operations together constitute a model for genome rearrangements that is often used for studying the evolution of the metazoan mitochondrial genomes [16].

Blanchette, Kunisawa, and Sankoff [22] also explored an idea of weighting each rearrangement based on its *length*, or the number of genes that it affects. This approach is based on the observed prevalence of short reversals in certain genomes [85, 35]. See Galvao, Baudet, and Dias [51] for a summary of the work on allowing only reversals of length at most 3. See Lintzmayer, Fertin and Dias [70] for a summary of the work on weighting reversals and transpositions based on their lengths.

Baudet, Dias, and Dias [9] proposed a model where reversals in bacterial genomes are weighted both based on their length and on the symmetry around the origin of replication. It was Ohlebusch, Abouelhoda, and Hockel [78] that first used this symmetry to provide a polynomial time algorithm for a variant of the reversal median problem. Their work was motivated by an observation that most of the reversals in some circular bacterial genomes are either centered around the origin/terminus of replication or involve a single gene [39]. As far as we are aware, this was the first time when external biological information of any kind was used in order to constraint a rearrangement scenario.

Over the last few years more work along these lines emerged. Biller, Guéguen, Knibbe, and Tannier [20] pointed out that the number of nucleotides in the intergenic regions, or their *lengths*, could be used as a biological constraint for genome rearrangements. Bulteau, Fertin, Jean and Tannier aimed to minimize the sum

of the lengths of insertions/deletions in the intergenic regions during a DCJ scenario [47, 27]. Similar ideas were later explored by Brito, Jean, Fertin, Oliveira, Dias, and Dias for reversals and transpositions of length at most 2 [79], and for reversals and transposition [26]. In the latter paper some possibilities for attributing different weights to reversals, transpositions and insertions/deletions were also explored.

Véron, Lemaitre, Gautier, Lacroix, and Sagot [99] suggested that the 3D structure of a genome is partially conserved between human and mouse. Physical proximity is already known to be one of the triggers for rearrangements in somatic cells [71], and Swenson and Blanchette [94] used Hi-C data to support this hypothesis in an evolutionary setting. Based on these observations our team aimed to minimize the number of DCJ rearrangements breaking regions distant in the 3D space [95, 91, 90, 89].

Only a tiny fraction of these weighted genome rearrangement problems have polynomial time algorithms. These include finding a minimum length scenario for reversals of length at most 2 [51, 10]. Also if reversal's weight is equal to its length, then a minimum weight reversal scenario for a binary string can be found in polynomial time as shown by Bender, Ge, He, Hu, Pinter, Skiena, and Swidan [10]. In addition to this, Bérard, Bergeron, Chauve, and Paul [11] introduced a specific family of gene clusters for which a minimum length preserving reversal scenario can be found in polynomial time. Other than that, all the weighted genome rearrangement problems with efficient algorithms of which we are aware are those that search for an optimal DCJ scenario among those that are of minimum length. These include a work on preserving DCJ by Bérard, Chateau, Chauve, Paul, and Tannier [12], on insertions and deletions in the intergenic regions by Bulteau, Fertin, and Tannier [27], and our own work on the physical locations of the intergenic regions [95, 90, 89].

1.4 A General Framework for Cost Constrained DCJ

We have seen that a parsimony criterion alone cannot provide a small enough subset of plausible rearrangement scenarios. Models for weighting genome rearrangements based on a number of biological constraints were explored with a hope that they might help with this issue. As presented in the previous section, this led to an accumulation of algorithmic work, however these models mostly exploited the combinatorial properties of the gene orders and rearrangement operations, and not the external biological constraints, such as those concerning the 3D structure of a genome, or fragility of certain loci that might be due to transcriptional activity or non-canonical DNA structures. In addition to this, only a handful of exact polynomial time algorithms for finding a minimum weight rearrangement scenario can be

found in the literature. However one line of work, that of weighted parsimonious DCJ scenarios, seemed promising to us and served as a starting point for our project.

In [27] and [95] two simple ways of assigning weights to DCJ rearrangements were introduced, while the weight of a DCJ scenario was defined to be the sum of the weights of its constituent rearrangements. In these papers genomes were supposed to be *single-copy*, meaning that they contain equal sets of genes, with a single occurrence of each gene. For both types of weights it was shown that finding a minimum weight scenario is NP-hard [47, 91], while a minimum weight scenario among the parsimonious ones can be found in polynomial time [27, 95]. As in the previous models for weighting genome rearrangements, these algorithms exploited the combinatorial properties of the models themselves and could not be easily generalized to be used with more sophisticated ways to weight DCJs.

A catalyst for our project was the work on sorting permutations with cost constrained transpositions by Farnoud and Milenkovic [41]. In the field of genome rearrangements a *transposition* is a rearrangement that swaps two adjacent regions of a chromosome, where a region can contain any number of genes. However in mathematics more broadly a *transposition* is a permutation that exchanges any two elements. It is the latter notion that we will use throughout this text. Farnoud and Milenkovic [41] allowed for an arbitrary cost function on transpositions and defined the cost of a transposition decomposition of a permutation to be the sum of the costs of its transpositions. Within this setting they showed that a minimum cost transposition decomposition among those of minimum length can be found in polynomial time regardless of the cost function. The ideas encountered in [41] encouraged us to aim for the following generalization of our work on weighting DCJs:

Project. *A framework for cost constrained genome rearrangements under a DCJ model within which a minimum cost parsimonious scenario between two single-copy genomes could be found in polynomial time for an arbitrary cost function.*

This means that our goal is not to come up with a particular mathematical model for cost constraining DCJs in a biologically meaningful way. But rather to provide guidelines for such a process, and ensure that no algorithmic work is needed if our guidelines are respected. The axes of our project became the following:

1. Efficiently explore the space of the parsimonious DCJ scenarios.
2. Augment a DCJ rearrangement with information concerning the intergenic regions that it cuts and joins, and define an arbitrary cost function on these augmented DCJs.
3. Efficiently search for a minimum cost scenario among the parsimonious ones.

These tasks are respectively treated in Chapter 3, Chapter 4, and Chapter 5. Our framework generalizes all three models for cost constraining DCJs of which we

are aware. These include previously mentioned work by our own team [95], Bulteau, Fertin, and Tannier [27], and Bérard, Chateau, Chauve, Paul, and Tannier [12]. However, as we discuss in Chapter 4, the latter model on preserving DCJs needs to be slightly modified for it to fit into our framework.

An important feature of our framework is that it enables us to easily combine different models for cost constraining genome rearrangement. This means, for example, that we can assign a cost to a DCJ simultaneously based on the locations of its breakpoint regions in the 3D space, their lengths and the number of the gene clusters that this DCJ preserves.

Another feature is that even if a polynomial time algorithm for finding a minimum cost parsimonious scenario is ensured only for single-copy genomes, we still provide an exact algorithm for genomes with multiple copies of genes and unequal gene content.

Finally, the true evolutionary scenario might be non-parsimonious, thus a long term goal is to move away from a parsimony criterion. To this end, it might be possible to avoid exploring the space of all the DCJ scenarios, as was previously done by Fertin, Jean, and Tannier [47] and our own team [91]. Statistical tools could be used instead to estimate an upper bound l for the length of the true evolutionary scenario [20, 2], and only the scenarios of length less than l could be explored. A number of our results are actually proved for non-parsimonious scenarios and provide a foundation for this future work.

In Chapter 2 we show that a DCJ can be interpreted as a graph transformation, which is a common approach in the field of genome rearrangements [48]. In the subsequent chapters we actually introduce a framework for cost constrained graph transformations, and not for cost constrained genome rearrangements. This general setting allows us to establish novel links between various sorting problems on permutations, strings, genomes and graphs.

1.5 Sorting Graphs with 2-breaks

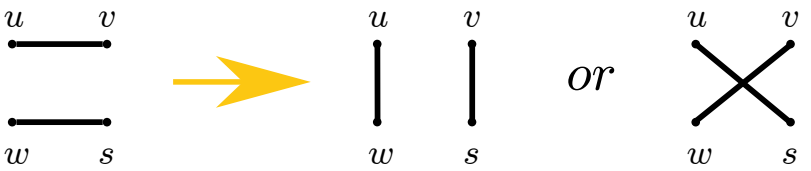


Figure 1.1: A 2-break transforms a pair of edges $\{u, v\}$ and $\{w, s\}$ of a multigraph into either $\{u, w\}$ and $\{v, s\}$, or $\{u, s\}$ and $\{v, w\}$. The term *2-break* was first proposed by Alekseyev and Pevzner [1].

Work on sorting permutations [41], strings [4] and genomes [55] with various

operations such as transpositions, interchanges and reversals relies heavily on graph theoretic techniques and data structures. In this work we show that a *2-break*, a graph transformation that swaps endpoints of two edges as illustrated in Figure 7.1, generalizes a number of different sorting operations. These include sorting permutations with transpositions [41], sorting strings with interchanges [4], sorting genomes with double cut and join (DCJ) rearrangements [102], and token swapping on graphs [23]. All of these problems have cost-constrained variants, which serves as motivation for our choice to study cost constrained 2-breaks.

A 2-break is known under many different names across the literature: (double edge) swap, (degree-preserving) rewiring, switch, shuffle, flip, checkerboard swap, and is the simplest way to transform a graph without changing its degree sequence [49]. Besides sorting, 2-breaks are used for modeling and analyzing various dynamic networks, notably peer-to-peer [43] and lightwave networks [19]. 2-breaks are also widely used for generating random graphs with an empirically relevant degree sequence [77], and for studying the *configuration model*, which is a uniform distribution over graphs with a specific degree sequence. See [49] for a recent survey of this line of work.

1.6 Outline

We start our work by introducing 2-breaks and a problem of finding a minimum length *2-break scenario* transforming one graph into another in Section 2.2. We proceed by showing that 2-breaks generalize various transformations of permutations, strings, genomes and graphs in Section 2.7.

In Chapter 2 we introduce some theory concerning genomes and permutations. We show that a DCJ scenario for a pair of genomes and a transposition decomposition of a permutation can be both seen as 2-break scenarios for their *breakpoint graphs*. Such observations have already been made before, however by combining them together we establish a novel result. We show that a parsimonious DCJ scenario for a pair of single-copy *co-tailed* genomes (these are genomes that share the sets of genes adjacent to the ends of their linear chromosomes) can be interpreted as a minimum length transposition decomposition of a permutation and vice versa. We conclude this chapter with a short discussion of how this link might help us with various median problems.

In Chapter 3 we concentrate on a parsimonious 2-break scenario for a graph, and demonstrate that it can be partitioned into parsimonious 2-break scenarios for the very basic subgraphs of the graph, that we call *circles*. We then proceed by showing that a parsimonious 2-break scenario for a circle can be partitioned into scenarios for smaller circles. This leads to a dynamic programming algorithm for exploring the space of the parsimonious 2-break scenarios for a circle and, due to the previous

observation, for any graph.

In Chapter 4 we introduce our framework for cost constraining 2-breaks that consists of four steps:

- Labeling vertices and edges of a graph.
- Allowing a 2-break to replace a pair of labeled edges with another pair of labeled edges.
- Allowing a label of an edge to be changed.
- Defining an arbitrary cost function φ on these new operations.

Within this setting we introduce the φ-Minimum Cost Parsimonious Scenario problem (φ-MCPS) for a labeled graph, and show that previous work on cost constrained DCJs [27, 95, 12] and cost constrained transposition decompositions [41] can be interpreted as φ-MCPS problems.

In Chapter 5 we provide an exact algorithm for the φ-MCPS problem. Its worst case time complexity for a genome breakpoint graph of a pair of single-copy genomes with n genes is $O(n^5L^4)$, where L is the number of edge labels allowed in the chosen model for cost constraining 2-breaks. We conclude this chapter by arguing that despite its elevated time complexity our algorithm remains of practical importance for the study of genome rearrangements.

Chapter 2

Linking 2-breaks on Graphs, Rearrangements on Genomes, and Transpositions on Permutations

2.1 Introduction

We start this chapter with a presentation of the theory of sorting graphs with 2-breaks. In Sections 2.3 and 2.4 we demonstrate that sorting genomes with double cut and join (DCJ) rearrangements and sorting permutations with transpositions can be both interpreted as sorting graphs with 2-breaks.

Some links between these two problems have already been explored by the people working on the algebraic rearrangement theory [73, 46], a recent survey of which is proposed by Bhatia, Feijão and Francis [18]. What is new here, is that we establish a bijection between the Minimum Length Transposition Decompositions (MLTDs) of a permutation and the parsimonious DCJ scenarios for a well chosen pair of genomes.

This bijection facilitates an exchange of ideas between two well studied problems. For example, the work of Farnoud and Milenkovic [41] on sorting permutations with cost constrained transpositions informed our work in Chapter 5 on sorting graphs with cost constrained 2-breaks. In its turn, our work generalizes that on the cost constrained transpositions, as it will be briefly discussed in Section 2.7. On the other hand, as we explain in Section 2.6, the complexity of the Swap Median Permutation problem [82] remains unknown, while the DCJ Median problem is known to be NP-hard [97]. The work on the latter problem might inform us on how the former could be approached.

2.2 Transforming Graphs with 2-breaks

2.2.1 A 2-break Scenario for a Graph

In this section we introduce the problem of sorting a 2-edge-colored graph with 2-breaks. This sorting problem is most often seen as transforming a source object (black edges) to the target one (gray edges), however here we will transform both black and gray edges into some intermediate set. This way to pose the problem does not change the length of a parsimonious 2-break scenario, however it will allow us to impose different cost constraints on black and gray edges in Chapter 4.

Definition 1 (2-edge-colored multigraph)**.** *Take a set V of vertices and two colors $\{black, gray\}$. An* edge *is an unordered pair of vertices. A* colored edge *is a pair of an edge and a color. A* 2-edge-colored multigraph *is an ordered pair (V, E), with E being a multiset of colored edges.*

Definition 2 (Eulerian graph and alternating cycle)**.** *A 2-edge-colored multigraph is* Eulerian *if its every vertex has equal black and gray degrees. A cycle is* alternating *if it is Eulerian.*

See Figure 2.1 a) for an example. All use of the word *graph* will be synonymous with *Eulerian 2-edge-colored multigraph*, and use of the word *cycle* will be synonymous with *alternating cycle*, unless specified otherwise.

Definition 3 (Terminal graph)**.** *A graph with equal multisets of black and gray edges is called* terminal*.*

Definition 4 (2-break)**.** *A 2-break transforms a pair of colored edges $(\{u, v\}, col)$ and $(\{w, s\}, col)$ into either $(\{u, w\}, col)$ and $(\{v, s\}, col)$, or $(\{u, s\}, col)$ and $(\{v, w\}, col)$. We denote the former of these transformations by $\tau = \Big(\big\{\{u, v\}, \{w, s\}\big\} \to \big\{\{u, w\}, \{v, s\}\big\}, col\Big)$. Take a graph G containing the colored edges $(\{u, v\}, col)$ and $(\{w, s\}, col)$, and a graph G' in which these edges were replaced with $(\{u, w\}, col)$ and $(\{v, s\}, col)$. We say that τ* transforms *G into G', and denote this transformation by $G \to G'$.*

Definition 5 (Vertices, edges and color of a 2-break)**.** *The* vertices *and* edges *of a 2-break $\tau = \Big(\big\{\{u, v\}, \{w, s\}\big\} \to \big\{\{u, w\}, \{v, s\}\big\}, col\Big)$, are respectively $\big\{u, v, w, s\big\}$ and $\big\{\{u, v\}, \{w, s\}, \{u, w\}, \{v, s\}\big\}$. Its* color *is col. We say that τ* replaces *colored edges $(\{u, v\}, col)$ and $(\{w, s\}, col)$ and* introduces *colored edges $(\{u, w\}, col)$ and $(\{v, s\}, col)$.*

Definition 6 (2-break scenario for a graph and its minimum length $d_{2b}(G)$)**.** *A 2-break* scenario *for a graph is a sequence of 2-breaks transforming it into a terminal graph. Denote the minimum length of a 2-break scenario for G by $d_{2b}(G)$.*

Observation 1. *If a 2-break scenario ρ for G contains a 2-break replacing a colored edge $(\{i, j\}, col)$, then either it is already present in G or ρ contains a 2-break introducing this labeled edge.*

Definition 7 (Black-2-break scenario and its minimum length $d_{b2b}(G)$)**.** *A black-2-break scenario is a 2-break scenario consisting entirely of black 2-breaks. Denote the minimum length of a black-2-break scenario for a graph G by $d_{b2b}(G)$.*

2.2.2 Maximum Alternating Edge-disjoint Cycle Decomposition

The problem of finding a parsimonious 2-break scenario for a graph is closely related to that of finding its Maximum Alternating Edge-disjoint Cycle Decomposition.

Definition 8 (Eulerian decomposition of a graph and its size $c(G)$)**.** *An* edge-disjoint decomposition *of a graph G is a set $\mathcal{H}$ of subgraphs of G whose edges partition the edges of G. $\mathcal{H}$ is an Eulerian decomposition (ED) if all of its subgraphs are Eulerian. $\mathcal{H}$ is a Maximum Alternating Edge-disjoint Cycle Decomposition (MAECD) if it has the most subgraphs among the EDs of G. Denote by $c(G)$ the number of subgraphs in an MAECD of G.*

Definition 9 (Simple cycle and circle)**.** *A graph is a* simple cycle *if its MAECD is of size 1. If in addition to that the black and gray degrees of its every vertex are equal to 1, then it is a* circle.

Observation 2. *An MAECD of a graph consists entirely of simple cycles due to maximality.*

See Figure 2.1 f)-g) for an example.

2.2.3 The Minimum Length of a 2-break Scenario

The problem of finding a parsimonious 2-break scenario has been treated in several unrelated settings using different terminologies. Bienstock and Günlük [19] provide a thorough analysis of the problem. They demonstrate that finding a minimum length black-2-break scenario is NP-hard due to the NP-hardness of finding an MAECD of a graph. They also provide a 7/4-approximation algorithm for finding its length.

Lemma 1 (Adapted from Bienstock and Günlük [19])**.** *The minimum length of a 2-break scenario for a graph G is $d_{2b}(G) = e(G) - c(G)$, where $e(G)$ is the number of colored edges in G divided by two.*

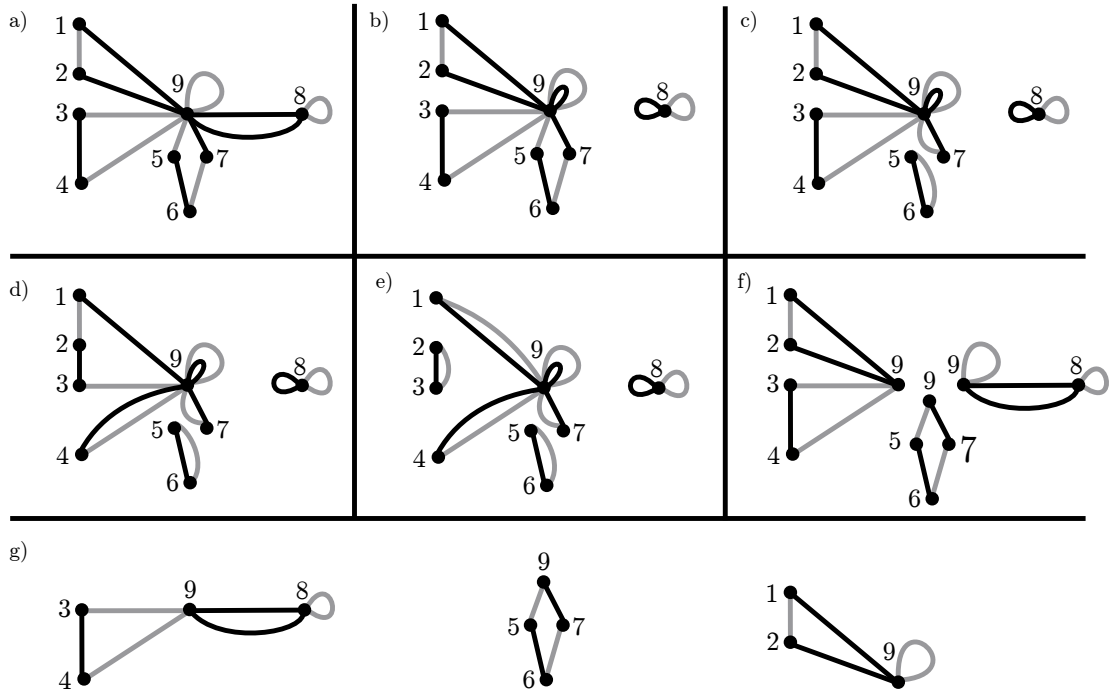


Figure 2.1: An example of an Eulerian 2-edge-colored multigraph G is presented in a). a) to e) depicts a parsimonious 2-break scenario transforming G into a terminal graph presented in e). The first two 2-breaks of the scenario are $\Big(\Big\{\{8,9\},\{8,9\}\Big\} \to \Big\{\{8,8\},\{9,9\}\Big\}, black\Big)$ and $\Big(\Big\{\{5,9\},\{6,7\}\Big\} \to \Big\{\{5,6\},\{7,9\}\Big\}, gray\Big)$. Two different Maximum Alternating Edge-disjoint Cycle Decompositions of G exist, they are depicted in f) and g). All the graphs depicted in g) are simple cycles. The one in the middle of g) is also a circle.

Proof. We start by showing that a 2-break increases the size of an MAECD by at most one. Take a 2-break transforming G into G' and an MAECD $\mathcal{H}'$ of G'. Remove from $\mathcal{H}'$ a subgraph or a pair of subgraphs containing the colored edges introduced by the 2-break, thus obtaining $\mathcal{H}$, a set of edge-disjoint Eulerian subgraphs of G. Remove them all from G to obtain its Eulerian subgraph H, and add it to $\mathcal{H}$ to obtain an Eulerian decomposition of G of size $c(G')$ or $c(G')-1$. A 2-break scenario transforms G into a terminal graph, that, by construction, has an MAECD of size $e(G)$. This means that $d_{2b}(G) \geq e(G) - c(G)$, as we need to increase the size of G's MAECD by $e(G) - c(G)$.

Now we show that there always exists a 2-break increasing the size of an MAECD by one. Take a non-terminal simple cycle S, its gray edge, and a pair of black edges incident to its endpoints. Denote them by $(\{u,w\}, gray)$, $(\{u,v\}, black)$, and $(\{w,s\}, black)$. A 2-break $\Big(\Big\{\{u,v\},\{w,s\}\Big\} \to \Big\{\{u,w\},\{v,s\}\Big\}, black\Big)$ provides a non-simple graph, as it contains colored edges $(\{u,w\}, gray)$ and $(\{u,w\}, black)$ forming a cycle. See Figure 2.2 for an example. This establishes that $d_{2b}(S) =$

$e(S) - 1$. Finally, take an MAECD $\mathcal{H}$ of G and sort its simple cycles one by one to obtain a 2-break scenario of length $e(G) - c(G)$. □

Corollary 1. *We have shown that there exists a black-2-break scenario of length* $e(G) - c(G)$*, thus* $d_{b2b}(G) = d_{2b}(G)$*.*

Corollary 2. *A prefix of length* m *of a parsimonious 2-break scenario for a circle transforms it into a vertex disjoint union of* $m + 1$ *circles.*

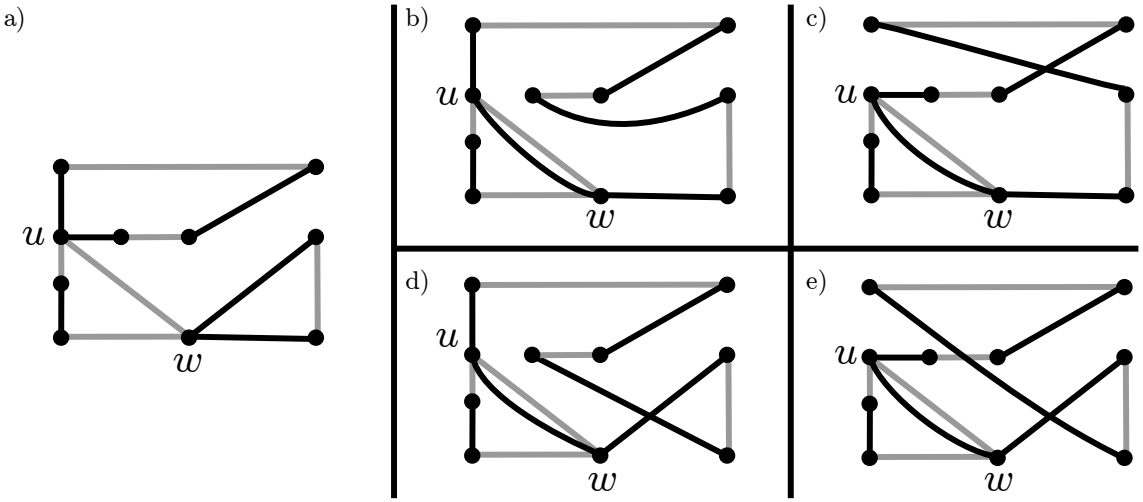


Figure 2.2: A simple cycle is depicted in a) with vertices u and w singled out. b)-d) depicts four possible 2-breaks that each yield a graph with an MAECD size equal to 2.

2.2.4 Equivalent 2-break Scenarios

In Chapter 4 we introduce cost constraints on the individual 2-breaks and define the cost of a scenario to be the sum of the costs of its individual moves. In this setting two 2-break scenarios containing the same 2-breaks, but possibly performed in different orders, have the same cost. This motivates the following definition of an equivalence relation between the 2-break scenarios.

Definition 10 (Equivalent 2-break scenarios)**.** *Two 2-break scenarios are* equivalent *if their multisets of 2-breaks are equal.*

In Chapter 3 we categorize the equivalence classes of the parsimonious 2-break scenarios.

2.3 Double Cut and Join Rearrangements are 2-breaks

2.3.1 Genomes

In this section we present the theory of sorting genomes by double cut and join (DCJ) rearrangements. We introduce the *breakpoint graph*, a graphical representation of a pair of genomes, and proceed by establishing a bijection between the DCJ scenarios for a pair of genomes and the 2-break scenarios for their breakpoint graph.

A *genome* consists of *chromosomes* that are two stranded linear or circular DNA molecules. In the field of genome rearrangements a chromosome is usually modeled as a circular or linear order of *directed genes* separated by *breakpoint* regions, where the *direction* of a gene indicates the strand of a chromosome that it belongs to [48]. In Figure 2.3 a) the tail of an arrow represents the *tail extremity*, and the head of an arrow represents the *head extremity* of a gene. A genome can be represented by its *adjacency set*. An *adjacency* is either *internal*: an unordered pair of the gene extremities that are adjacent on a chromosome, or *external*: a single gene extremity adjacent to one of the two ends of a linear chromosome.

Denote the set of genes (or *syntenic blocks*) by $V = \{1, \ldots, n\}$, and the sets of their head an tail extremities by $V_h = \{1_h, \ldots, n_h\}$ and $V_t = \{1_t, \ldots, n_t\}$. In what follows we suppose that genomes share the same multiset of genes, however the proposed methods also apply for the genomes with different gene content when *ghost* adjacencies are introduced to account for the missing genes, as proposed by Yancopoulos and Friedberg [103]. See Figure 2.3 for an example.

2.3.2 Double Cut and Join and the Adjacency Graph

We use *double cut and join* to model genome rearrangements.

Definition 11 (Double cut and join (DCJ) [102])**.** *A DCJ transforms the adjacencies of a genome in one of the four following ways:*

1. $\big\{\{a,b\},\{c,d\}\big\} \to \big\{\{a,c\},\{b,d\}\big\}$,
2. $\big\{\{a,b\},\{c\}\big\} \to \big\{\{a,c\},\{b\}\big\}$,
3. $\big\{\{a\},\{b\}\big\} \to \big\{\{a,b\}\big\}$,
4. $\big\{\{a,b\}\big\} \to \big\{\{a\},\{b\}\big\}$,

with $a, b, c, d \in V_h \cup V_t$*. Take a genome* A *containing the adjacencies* $\{a,b\}$ *and* $\{c,d\}$*, and a genome* A' *in which these adjacencies were replaced with* $\{a,c\}$ *and*

$\{b,d\}$. *We say that a DCJ* $\Big(\big\{\{a,b\},\{c,d\}\big\} \to \big\{\{a,c\},\{b,d\}\big\}, A\Big)$ transforms A *into* A', *and denote this transformation by* $A \to A'$.

We will mostly be interested in the genomes containing a single copy of each gene. Such *single copy genomes* are uniquely represented by their sets of adjacencies, which is not the case for the genomes in general. All the work on DCJ of which we are aware operates on the adjacency sets and deals with the following notions of DCJ scenario and DCJ distance.

Definition 12. *A DCJ* scenario *is a sequence of DCJs transforming two genomes* A *and* B *into two genomes with equal sets of adjacencies. Denote by* $d_{aDCJ}(A,B)$ *the minimum length of such a scenario for genomes* A *and* B.

Observation 3. $d_{aDCJ}(A,B)$ *is called* edit distance *in [86] and* generalized DCJ distance *in [88]. See Figure 2.3 for an example of non-equal genomes with* d_{aDCJ} *equal to 0.*

A more sophisticated DCJ distance introduced in Definition 13 that takes the structure of the chromosomes into account remains to be studied in the future.

Definition 13. *A* genome DCJ scenario *is a sequence of DCJs transforming two genomes* A *and* B *into two equal genomes. Denote by* $d_{gDCJ}(A,B)$ *the minimum length of such a scenario for genomes* A *and* B.

Observation 4. *For any pair of genomes we have* $d_{aDCJ}(A,B) \leq d_{gDCJ}(A,B)$, *while* $d_{aDCJ}(A,B) = d_{gDCJ}(A,B)$ *if* A *and* B *are single copy genomes. See Figure 2.3 for an example of genomes with unequal values of* d_{aDCJ} *and* d_{gDCJ}.

The *adjacency graph* introduced by Bergeron, Mixtacki, and Stoye [15] has been widely used to represent a pair of genomes and study their DCJ scenarios. See Figure 2.4 for an example.

Definition 14 (Adjacency graph)**.** *The graph* $AG(A,B)$ *of two genomes is the bipartite graph whose vertices are the adjacencies of* A *and* B. *There is an edge (respectively two edges) between the adjacencies if they share a gene extremity (respectively two gene extremities).*

With the help of an adjacency graph the following results were established.

Lemma 2 (Bergeron, Mixtacki, and Stoye [15])**.** $d_{aDCJ}(A,B)$ *for the single copy genomes is equal to* $n-(C+I/2)$, *where* n *is the number of genes,* C *is the number of cycles of* $AG(A,B)$ *and* I *is the number of odd length paths among the connected components of* $AG(A,B)$.

Lemma 3 (Shao, and Lin [86])**.** *Computing $d_{aDCJ}(A, B)$ for the non-single copy genomes is NP-hard. Denote by $\mathcal{D}$ the set of* vertex-disjoint decompositions *of $AG(A, B)$ into cycles and paths. $d_{aDCJ}(A, B) = n - \max_{D\in\mathcal{D}}(c_D + o_D/2)$, where c_D is the number of cycles and o_D is the number of odd length paths in a decomposition D.*

These results can be seen as corollaries of Theorem 1 that we establish in what follows.

2.3.3 The Breakpoint Graph

The adjacency graph conveniently represents genomes, however we will only be interested in their adjacency sets. We find that in this setting the *breakpoint graph* [7], being more concise, is much easier to work with. Here we use a variant of the breakpoint graph very similar to the one introduced by Alekseyev and Pevzner [1]. See Figure 2.4 and Figure 2.3 for examples.

Definition 15 (Genome breakpoint graph)**.** *$G(A, B)$ for two genomes A and B is a 2-edge-colored multigraph on vertices $V_h \cup V_t \cup \{\circ\}$. For every internal adjacency $\{a, b\} \in A$ (respectively B) there is a colored edge $(\{a, b\}, black)$ (respectively $(\{a, b\}, gray)$) in $G(A, B)$, and for every external adjacency $\{a\} \in A$ (respectively B) there is a colored edge $(\{a, \circ\}, black)$ (respectively $(\{a, \circ\}, gray)$) in $G(A, B)$. There is also a number of loops $(\{\circ, \circ\}, black)$ and $(\{\circ, \circ\}, gray)$ in $G(A, B)$ ensuring that the black and gray degrees of $\circ$ are equal to $2n$.*

Observation 5. *The genome breakpoint graph is Eulerian.*

Definition 16 (AA/BB paths of $AG(A, B)$)**.** *Take a connected component of the adjacency graph $AG(A, B)$ that is an even length path. If it starts in A (respectively B), then it is an AA* (respectively BB) path *of $AG(A, B)$.*

Definition 17 (AA/BB paths of $G(A, B)$)**.** *Take a connected non Eulerian subgraph H of $G(A, B)$ in which the black and gray degrees of every vertex different from $\circ$ are equal to 1. If the black and gray degrees of $\circ$ are respectively equal to 2 and 0, then H is an AA* path *of $G(A, B)$. If these degrees are respectively equal to 0 and 2, then H is a BB* path *of $G(A, B)$.*

See Figure 2.4 for an example of AA and BB paths of $AG(A, B)$ and $G(A, B)$.

Observation 6. *Every vertex of the adjacency graph $AG(A, B)$ corresponds to an edge of the genome breakpoint graph $G(A, B)$. This way, for every path in $AG(A, B)$ there exists a corresponding path in $G(A, B)$.*

Using Lemma 2 we establish a following lemma.

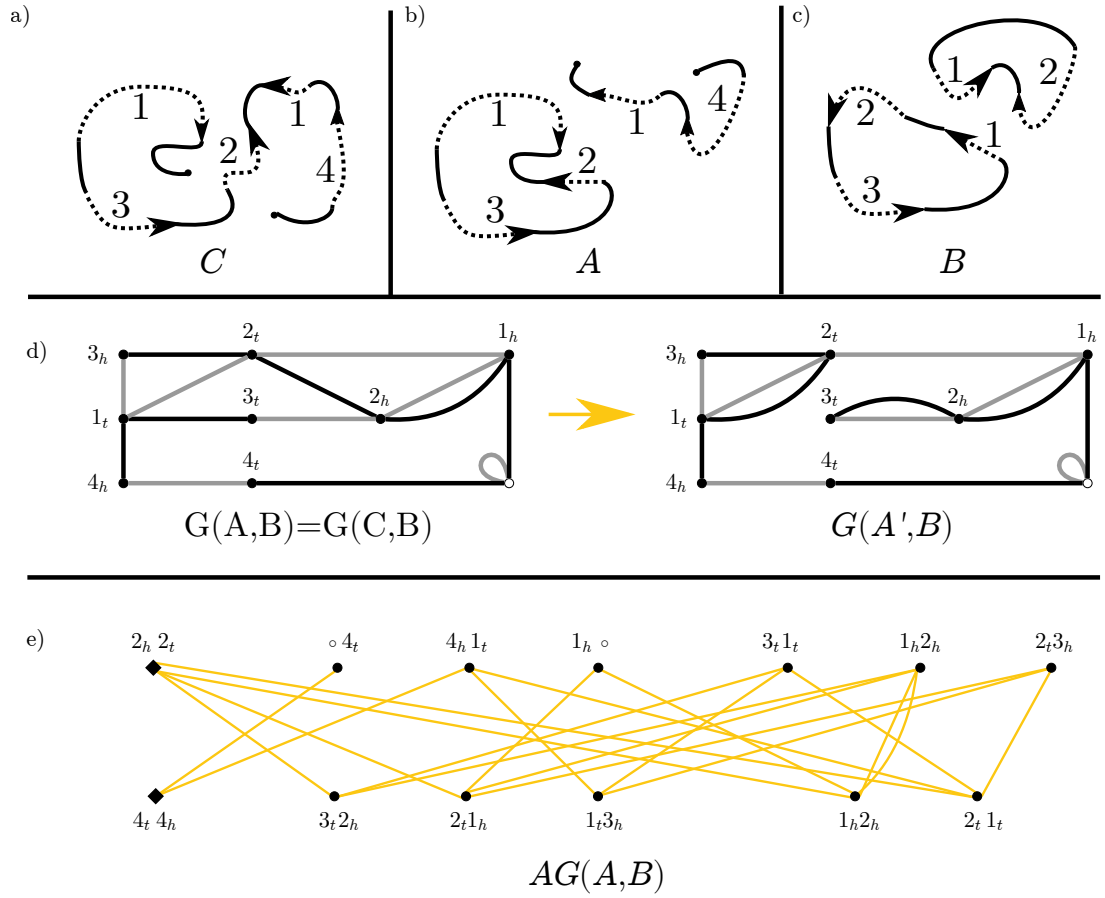
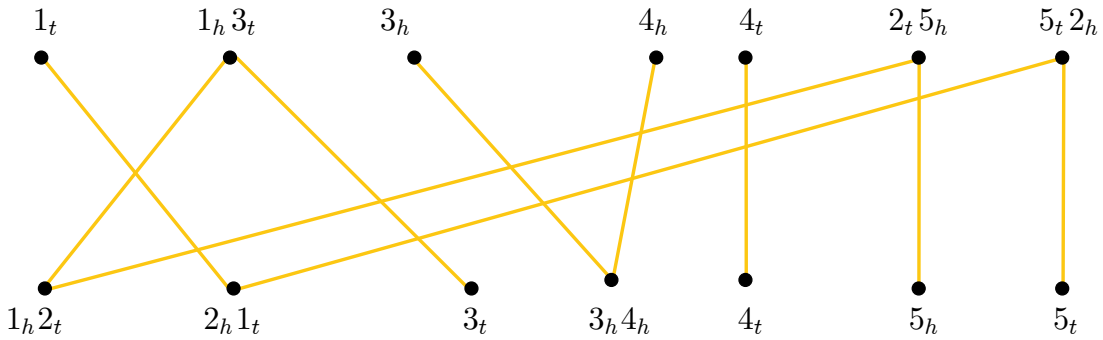
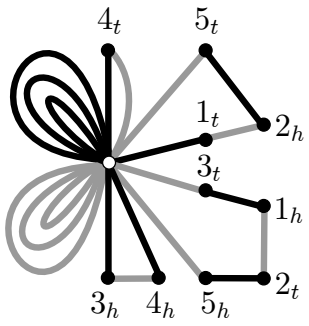


Figure 2.3: Three genomes $C = \{(\{4_t\},\{4_h,1_t\},\{1_h,2_h\},\{2_t,3_h\},\{3_t,1_t\},\{1_h\})\}$, $A = \{(\{3_t,1_t\},\{1_h,2_h\},\{2_t,3_h\}),\ (\{4_t\},\{4_h,1_t\},\{1_h\})\}$, and $B = \{(\{1_h,2_h\},\{2_t,1_t\}),\ (\{3_t,2_h\},\{2_t,1_h\},\{1_t,3_h\})\}$ are depicted in a)-c). C consists of a linear chromosome, A consists of a linear and a circular chromosome, while B consists of two circular chromosomes. The sets of adjacencies of C and A are the same, thus $d_{aDCJ}(A,C) = 0$, while $d_{gDCJ}(A,C) = 1$. Take a genome $A' = \{(\{3_t,2_h\},\{2_t,1_t\},\{1_h,2_h\},\{2_t,3_h\}),\ (\{4_t\},\{4_h,1_t\},\{1_h\})\}$. The breakpoint graphs $G(A,B)$ and $G(A',B)$ are depicted in d). The vertex $\circ$, representing the ends of the linear chromosomes, is colored white. Extra colored edges are added for the missing genes (*e.g.* $(\{2_t,2_h\},black)$ and $(\{4_h,4_t\},gray)$), called *ghost adjacencies* in [86]. The operation transforming A to A' is an insertion of gene 2. It corresponds to the 2-break $G(A,B) \to G(A',B)$. In e) the adjacency graph $AG(A,B)$ is depicted with diamonds indicating the ghost adjacencies.



$AG(A,B)$



$G(A,B)$

Figure 2.4: The adjacency and breakpoint graphs $AG(A, B)$ and $G(A, B)$ of the single copy genomes are depicted, with $A = \{(\{1_t\}, \{1_h, 3_t\}, \{3_h\}), (\{4_h\}, \{4_t\}), (\{2_t, 5_h\}, \{5_t, 2_h\})\}$ and $B = \{(\{1_h, 2_t\}, \{2_h, 1_t\}), (\{3_t\}, \{3_h, 4_t\}, \{4_h\}), (\{5_h\}, \{5_t\})\}$ Both A and B contain a circular and 2 linear chromosomes. $AG(A, B)$ contains two odd paths, an AA and a BB path. The odd path $(\{1_t\}, \{2_h, 1_t\}, \{5_t, 2_h\}, \{5_t\})$ of $AG(A, B)$, corresponds to the cycle $(\circ, 1_t, 2_h, 5_t, \circ)$ of $G(A, B)$. The AA path $(\{3_h\}, \{3_h, 4_h\}, \{4_h\})$ of $AG(A, B)$ corresponds to the AA path $(\circ, 3_h, 4_h, \circ)$ of $G(A, B)$. The BB path $(\{3_t\}, \{3_t, 1_h\}, \{1_h, 2_t\}, \{2_t, 5_h\}, \{5_h\})$ of $AG(A, B)$ corresponds to the BB path $(\circ, 3_t, 1_h, 2_t, 5_h, \circ)$ of $G(A, B)$.

Lemma 4. *For single copy genomes we have* $d_{aDCJ}(A, B) = d_{2b}(G(A, B))$.

Proof. Denote by C the number of cycles, and by I the number of odd paths among the connected components of $AG(A, B)$. There are C circles among the connected components of $G(A, B)$. There are also I subgraphs of $G(A, B)$ that are circles containing the vertex $\circ$. The rest of the simple cycles in $G(A, B)$ contain a pair of an AA and a BB path, and $G(A, B)$ contains $n - I/2$ AA and BB paths in total, thus $c(G(A, B)) = C + I + n - I/2$. By summing up the degrees of the vertices and dividing them by two we obtain that $e(G) = 2n$, thus $d_{2b}(G(A, B)) = e(G(A, B)) - c(G(A, B)) = n - C - I/2$ due to Lemma 1. Due to Lemma 2, $d_{aDCJ}(A, B) = n - C - I/2$, and thus $d_{2b}(G(A, B)) = d_{aDCJ}(A, B)$. □

In what follows we establish that the equality $d_{2b}(G(A, B)) = d_{aDCJ}(A, B)$ stays valid for the non-single copy genomes. The breakpoint graph $G(A, B)$ is defined in such a way, that for a DCJ $A \to A'$, the corresponding graph transformation $G(A, B) \to G(A', B)$ is a black 2-break. In addition to that, for a black 2-break $G(A, B) \to G'$ there exists a genome A' such that the transformation $A \to A'$ is a DCJ and $G(A', B) = G'$.

Definition 18 (DCJ of a 2-break and 2-break of a DCJ)**.** *Take a DCJ* δ *transforming adjacencies of either genome* A *or* B*. The four DCJ types presented in Definition 11 correspond to the four following 2-breaks in the same order:*

1. $\Big(\big\{\{a, b\}, \{c, d\}\big\} \to \big\{\{a, c\}, \{b, d\}\big\}, col\Big)$,
2. $\Big(\big\{\{a, b\}, \{c, \circ\}\big\} \to \big\{\{a, c\}, \{b, \circ\}\big\}, col\Big)$,
3. $\Big(\big\{\{a, \circ\}, \{b, \circ\}\big\} \to \big\{\{a, b\}, \{\circ, \circ\}\big\}, col\Big)$,
4. $\Big(\big\{\{a, b\}, \{\circ, \circ\}\big\} \to \big\{\{a, \circ\}, \{b, \circ\}\big\}, col\Big)$,

with $col = black$ *if* δ *transforms* A *and* $col = gray$ *if* δ *transforms* B*. This correspondence is used to define the notions of the* DCJ of a 2-break *and the* 2-break of a DCJ*. For example, we say that* $\Big(\big\{\{a, b\}, \{\circ, \circ\}\big\} \to \big\{\{a, \circ\}, \{b, \circ\}\big\}, black\Big)$ *is a 2-break* $\tau(\delta)$ *of a DCJ* $\delta = \Big(\big\{\{a, b\}\big\} \to \big\{\{a\}, \{b\}\big\}, A\Big)$*, and that* $\Big(\big\{\{a\}, \{b\}\big\} \to \big\{\{a, b\}\big\}, B\Big)$ *is a DCJ* $\delta(\tau)$ *of a 2-break* $\tau = \Big(\big\{\{a, \circ\}, \{b, \circ\}\big\} \to \big\{\{a, b\}, \{\circ, \circ\}\big\}, gray\Big)$.

Theorem 1. *Take a DCJ scenario* $\Delta = (\delta_1, \ldots, \delta_m)$ *for genomes* A *and* B*.* $\rho(\Delta) = (\tau(\delta_1), \ldots, \tau(\delta_m))$ *is a 2-break scenario for* $G(A, B)$*. Take a 2-break scenario* $\rho = (\tau_1, \ldots, \tau_m)$ *for* $G(A, B)$*.* $\Delta(\rho) = (\delta(\tau_1), \ldots, \delta(\tau_m))$ *is a DCJ scenario for* A *and* B*.*

Proof. Take a 2-break τ of color *col* transforming $G(A, B)$ to $G(A, B)'$. If *col* is black, then $\delta(\tau)$ can be performed on A to obtain a genome A' such that $G(A, B)' =$

$G(A', B)$. If *col* is gray, then $\delta(\tau)$ can be performed on B to obtain a genome B' such that $G(A, B)' = G(A, B')$. This means that $\Delta(\rho)$ transforms A and B into genomes $\bar{A}$ and $\bar{B}$, such that $G(\bar{A}, \bar{B})$ is a terminal graph. This implies that the sets of adjacencies of $\bar{A}$ and $\bar{B}$ are equal, and thus $\Delta(\rho)$ is a DCJ scenario for A and B.

Now take a DCJ δ transforming A into some genome A'. If δ is among the first three types in Definition 11, then $\tau(\delta)$ can be performed on $G(A, B)$ to obtain $G(A', B)$. The only non trivial case is if $\delta = \Big(\big\{\{a, b\}\big\} \to \big\{\{a\}, \{b\}\big\}, A\Big)$ for some extremities a and b. In this case $\tau(\delta) = \Big(\big\{\{a, b\}, \{\circ, \circ\}\big\} \to \big\{\{a, \circ\}, \{b, \circ\}\big\}, black\Big)$. In order for $\tau(\delta)$ to be a 2-break for $G(A, B)$, it must contain a self loop $(\{\circ, \circ\}, black)$. However such a loop is present, since A contains an internal adjacency $\{a, b\}$, thus it contains at most $2n - 2$ external adjacencies and the black degree of $\circ$ in $G(A, B)$ is $2n$ by construction. This means that a 2-break $\tau(\delta)$ can be performed on $G(A, B)$ to obtain $G(A', B)$. An analogous observation stays valid for a DCJ δ transforming genome B. This means that $\rho(\Delta)$ transforms $G(A, B)$ into a graph $G(C, C)$ for some genome C, and this graph is terminal by construction. Thus $\rho(\Delta)$ is a 2-break scenario for $G(A, B)$. □

Corollary 3. *For genomes A and B one has $d_{aDCJ}(A, B) = d_{2b}(G(A, B))$.*

2.3.4 Unsigned DCJs

Due to experimental limitations the directions (also known as *strandedness* or *signs*) of the genes might remain unknown. In such a case a chromosome can still be modeled as a circular or linear order of unsigned genes. For example a signed genome $A_s = \big\{(\{3_t, 1_t\}, \{1_h, 2_h\}, \{2_t, 3_h\}), (\{4_t\}, \{4_h, 1_t\}, \{1_h\})\big\}$ can be interpreted as an unsigned one $A_u = \big\{(\{3, 1\}, \{1, 2\}, \{2, 3\}), (\{4\}, \{4, 1\}, \{1\})\big\}$. The definitions of DCJ and the breakpoint graph remain valid for genomes containing unsigned genes, and so does Theorem 1. Chen [31] previously established this theorem for unsigned genomes using a slightly less general representation of a genome as a signed permutation.

2.4 Transpositions are 2-breaks

We start by introducing the basic notions of the theory of sorting permutations with *transpositions*. In the field of genome rearrangements a transposition is a rearrangement that swaps two adjacent regions of a chromosome, where a region can contain any number of genes. However in mathematics more broadly a transposition is a permutation that exchanges any two elements. It is the latter notion that we will use throughout this work. A *permutation* σ of $V = \{1, \ldots, n\}$ is a bijection $\sigma : V \to V$. Denote the set of permutations of V by $\mathbb{S}_n$. The *identity* permutation, which maps

every element onto itself, is denoted by id. The *product* of two permutations σ_1 and σ_2 is a permutation $\sigma_1\sigma_2$, such that $\sigma_1\sigma_2(i) = \sigma_1(\sigma_2(i))$ for any $i \in V$. Every permutation σ has an inverse σ^{-1}, such that $\sigma\sigma^{-1} = id$. An *orbit* of a permutation σ on $x \in V$ is the set $\{\sigma^n(x)|n \in \mathbb{Z}\}$. A permutation is a *cycle* if it has at most one orbit of size larger than 1. Denote the number of the disjoint orbits of a permutation σ by $c(\sigma)$. Such a permutation can be written as a product of $c(\sigma)$ disjoint cycles. A *transposition* is a cycle of length 2. Every permutation can be written as a product of transpositions, the minimum number of transpositions in such a product is denoted by $d_{Cayley}(\sigma, id)$. A sequence of transpositions $(\pi_1, \ldots, \pi_m)$, for which $\pi_m \ldots \pi_1\sigma = id$, with $m = d_{Cayley}(\sigma, id)$, is called a Minimum Length Transposition Decomposition (MLTD) of σ. The *Cayley distance* $d_{Cayley}(\sigma_1, \sigma_2)$ between two permutations is defined to be $d_{Cayley}(\sigma_2^{-1}\sigma_1, id)$.

Lemma 5 (Cayley [30])**.** *For every permutation σ on n elements $d_{Cayley}(\sigma, id) = n - c(\sigma)$.*

We proceed by introducing a graphical representation of permutations, similar to the breakpoint graph of the genomes, and by showing that there is a bijection between the parsimonious 2-break scenarios for that graph and the MLTDs of the permutations.

Definition 19 (Permutation breakpoint graph)**.** *For a pair of permutations (σ_1, σ_2) define a graph $H(\sigma_1, \sigma_2)$ on vertices $V_t = \{1_t, \ldots, n_t\}$ and $V_h = \{1_h, \ldots, n_h\}$ with black edges $\{(i_h, \sigma_1(i)_t)|i \in V\}$ and gray edges $\{(i_h, \sigma_2(i)_t)|i \in V\}$.*

Definition 20 (Positive circular graph)**.** *A graph is* circular *if the black and gray degrees of its vertices are equal to 1, which means that all of its connected components are circles. A graph is* positive *if it is a bipartite graph on vertices $V_h = \{1_h, \ldots, n_h\}$ and $V_t = \{1_t, \ldots, n_t\}$, with V_h and V_t being independent sets.*

Observation 7. *$H(\sigma_1, \sigma_2)$ is a positive circular graph.*

See Figure 2.5 b) for an example with $\sigma_1 = (123456)$ and $\sigma_2 = id$.

Lemma 6. *For a positive circular graph H there exist unique permutations σ_1 and σ_2 such that $H(\sigma_1, \sigma_2) = H$. If H is terminal, then $\sigma_1 = \sigma_2$.*

Proof. For every $i \in \{1, \ldots, n\}$ there exist unique $k, l \in \{1, \ldots, n\}$ such that H contains colored edges $(\{i_h, k_t\}, black)$ and $(\{i_h, l_t\}, gray)$. We define $\sigma_1(i) = k$ and $\sigma_2(i) = l$ to obtained the required permutations. □

Definition 21 (Colored transposition)**.** *A colored transposition is a pair of a transposition and a color $col \in \{black, gray\}$. A color indicates whether the transposition applies to σ_1 or σ_2.*

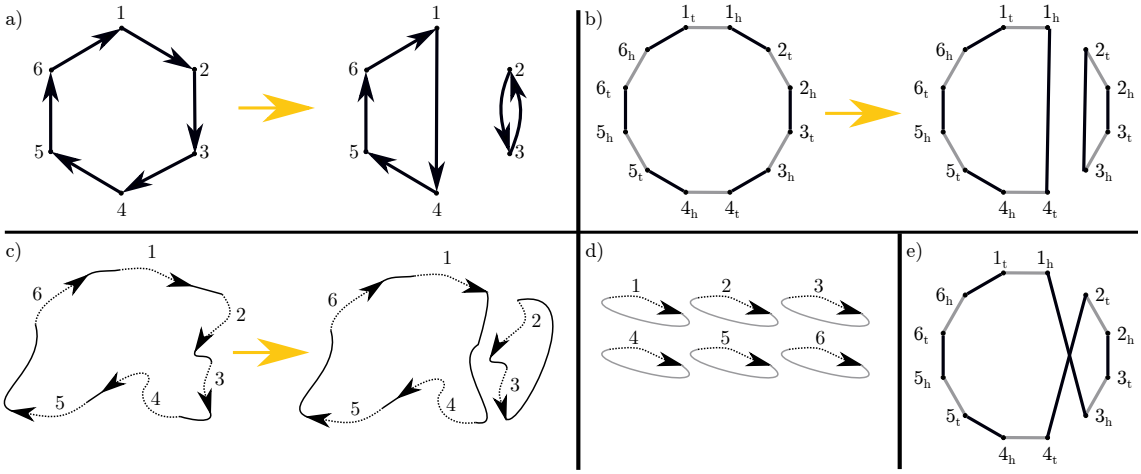


Figure 2.5: We provide an example for a permutation $\sigma = (123456)$ and a transposition $\pi = (24)$. Their product $\pi\sigma$ can be expressed as a product of two disjoint cycles $\pi\sigma = (1456)(23)$. A graphical representation of a permutation as a functional digraph $D(\sigma) = (V, \{(i, \sigma(i)) | i \in V\})$ is depicted in a) for $D(\sigma)$ and $D(\pi\sigma)$. The permutation breakpoint graphs $H(\sigma, id)$ and $H(\pi\sigma, id)$ depicted in b) differ by a 2-break $\Big(\big\{\{1_h, 2_t\}, \{3_h, 4_t\}\big\} \rightarrow \big\{\{1_h, 4_t\}, \{3_h, 2_t\}\big\}, black\Big)$. Two positive genomes (see Definition 28) A and A' are depicted in c). The DCJ transforming one into another is a circular excision. A positive genome B is depicted in d). $H(\sigma, id)$ in b) is an internal genome breakpoint graph (see Definition 27) of A and B, while $H(\pi\sigma, id)$ in b) is that of A' and B. A graph H' that can be obtained from $H(A, B)$ via a non preserving 2-break (see Definition 23) $\big\{\{1_h, 2_t\}, \{3_h, 4_t\}\big\} \rightarrow \big\{\{1_h, 3_h\}, \{2_t, 4_t\}\big\}$ is presented in e). This 2-break does not belong to any parsimonious 2-break scenario for $H(A, B)$. On a side note, in [46] a genome A gets assigned permutations $\pi_{chr} = (1_t2_t3_t4_t5_t6_t)(6_h5_h4_h3_h2_h1_h)$ and $\pi_{adj} = (1_h2_t)(2_h3_t)(3_h4_t)(4_h5_t)(5_h6_t)(6_h1_t)$, while the DCJ transforming A into A' is interpreted as a pair of transpositions $(1_h3_h)(2_t4_t)$. See Section 2.8 for a quick explanation.

Definition 22 (2-break $\tau(\pi, col)$ of a colored transposition)**.** *Take a transposition $\pi = (ij)$, a color col and a graph $H(\sigma_1, \sigma_2)$. Denote the unique edges of color col in $H(\sigma_1, \sigma_2)$ incident to vertices i_t and j_t by $(\{k_h, i_t\}, col)$ and $(\{l_h, j_t\}, col)$. The 2-break $\tau(\pi, col)$ is $\Big(\big\{\{k_h, i_t\}, \{l_h, j_t\}\big\} \rightarrow \big\{\{k_h, j_t\}, \{l_h, i_t\}\big\}, col\Big)$.*

Observation 8. *A 2-break $\tau(\pi, col)$ transforms $H(\sigma_1, \sigma_2)$ into $H(\pi\sigma_1, \sigma_2)$ if col is black and $H(\sigma_1, \pi\sigma_2)$ otherwise.*

Definition 23 (Preserving 2-break)**.** *A 2-break for a positive circular graph is* preserving*, if it transforms it into another positive circular graph.*

Definition 24 (Transposition $\pi(\tau)$ of a preserving 2-break τ)**.** *Exactly two vertices of a preserving 2-break τ are in V_t, denote them by i_t and j_t. The transposition $\pi(\tau)$ is (ij).*

Observation 9. *Take a preserving 2-break τ for $H(\sigma_1, \sigma_2)$. τ transforms $H(\sigma_1, \sigma_2)$ into $H(\pi(\tau)\sigma_1, \sigma_2)$ if it is black, into $H(\sigma_1, \pi(\tau)\sigma_2)$ otherwise.*

Lemma 7. *All the 2-breaks in a parsimonious 2-break scenario for a positive circular graph are preserving.*

Proof. It is enough to show that the first 2-break τ of a parsimonious 2-break scenario for a positive circular graph is preserving. τ replaces two colored edges of the same circle and transforms it into a union of vertex disjoint circles due to Corollary 2. Figure 2.5 b) and e) illustrate that the obtained circular graph is positive. □

Lemma 8. *Take an MLTD $(\pi_1, \ldots, \pi_m)$ of σ. $(\tau(\pi_1, black), \ldots, \tau(\pi_m, black))$ is a parsimonious black-2-break scenario for $H(\sigma, id)$. Take a parsimonious black-2-break scenario $(\tau_1, \ldots, \tau_l)$ for $H(\sigma, id)$. $(\pi(\tau_1), \ldots, \pi(t_l))$ is an MLTD of σ.*

Proof. Take a Minimum Length Transposition Decomposition $(\pi_1, \ldots, \pi_m)$ of σ. Due to Observation 8, a 2-break $\tau(\pi_i, black)$ transforms $H(\pi_{i-1} \ldots \pi_1\sigma, id)$ to $H(\pi_i\pi_{i-1} \ldots \pi_1\sigma, id)$, and thus $\rho = (\tau(\pi_1, black), \ldots, \tau(\pi_m, black))$ transforms $H(\sigma, id)$ to $H(\pi_m \ldots \pi_1\sigma, id) = H(id, id)$, which is a terminal graph. This means that ρ is a black-2-break scenario for $H(\sigma, id)$ and establishes that $m \geq l$.

Now take a parsimonious black-2-break scenario $(\tau_1, \ldots, \tau_l)$ transforming $H(\sigma, id)$ into a terminal graph $\overline{H(\sigma, id)}$. Due to Lemma 7, all the 2-breaks in the scenario are preserving. Due to Observation 9, τ_i transforms $H(\pi(\tau_{i-1}) \ldots \pi(\tau_1)\sigma, id)$ to $H(\pi(\tau_i)\pi(\tau_{i-1}) \ldots \pi(\tau_1)\sigma_1, \sigma_2)$, thus $\overline{H(\sigma, id)} = H(\pi(\tau_l) \ldots \pi(\tau_1)\sigma, id)$. Due to Lemma 6, $\pi(\tau_l) \ldots \pi(\tau_1)\sigma = id$, thus $(\pi(\tau_1), \ldots, \pi(\tau_l))$ is a transposition decomposition of σ. This establishes that $l \geq m$, and thus $m = l$, which means that $(\tau(\pi_1, black), \ldots, \tau(\pi_m, black))$ is a parsimonious black-2-break scenario for $H(\sigma, id)$ and $(\pi(\tau_1), \ldots, \pi(\tau_l))$ is an MLTD of σ. □

Lemma 8 can be easily generalized using the following definition of an MLTD of a pair of permutations.

Definition 25 (Shuffle)**.** *A* shuffle *of two sequences ρ_1 and ρ_2 is a sequence that can be partitioned into the sub-sequences equal to ρ_1 and ρ_2.*

Example 1. *$(b_1, b_2, a_1, b_3, a_2, a_3, b_4)$ is shuffle of (a_1, a_2, a_3) and (b_1, b_2, b_3, b_4).*

Definition 26 (MLTD of permutations)**.** *Take two sequences of colored transpositions $T_1 = ((\pi^1_1, black), \ldots, (\pi^1_{m_1}, black))$ and $T_2 = ((\pi^2_1, gray), \ldots, (\pi^2_{m_2}, gray))$, such that $\pi^1_{m_1} \ldots \pi^1_1\sigma_1 = \pi^2_{m_2} \ldots \pi^2_1\sigma_2$ and $m_1 + m_2 = d_{Cayley}(\sigma_1, \sigma_2)$. An MLTD of σ_1 and σ_2 is a shuffle of T_1 and T_2.*

Essentially the same proof as that of Lemma 8, allows us to establish a bijection between the parsimonious 2-break scenarios for $H(\sigma_1, \sigma_2)$ and the MLTDs of σ_1 and σ_2.

Theorem 2. *Take an MLTD* $((\pi_1, col_1), \ldots, (\pi_m, col_m))$ *of* σ_1 *and* σ_2*. A sequence of 2-breaks* $(\tau(\pi_1, col_1), \ldots, \tau(\pi_m, col_m))$ *is a parsimonious scenario for* $H(\sigma_1, \sigma_2)$*. Take a parsimonious 2-break scenario* $(\tau_1, \ldots, \tau_m)$ *for* $H(\sigma_1, \sigma_2)$*, with* $(col_1, \ldots, col_m)$ *being the colors of these 2-breaks. A sequence* $((\pi(\tau_1), col_1), \ldots, (\pi(t_m), col_m))$ *is an MLTD of* σ_1 *and* σ_2*.*

2.5 A Parsimonious DCJ Scenario for Co-tailed Single Copy Genomes is a Parsimonious 2-break Scenario for a Circular Positive Graph

2.5.1 Positive Genomes

We want to establish a link between the DCJ scenarios for single copy genomes and the transposition decompositions of permutations. It is not an immediate task due to the differences between the genome and permutation breakpoint graphs.

- The breakpoint graph of the single copy genomes is not circular, as it contains a vertex $\circ$ with the black and gray degrees greater than one.
- The genome breakpoint graph is not necessarily positive (see Definition 20).

In what follows we show how to overcome these two differences and interpret a parsimonious DCJ scenario as an MLTD and vice versa.

Definition 27 (Internal genome breakpoint graph)**.** *Take the genome breakpoint graph of two genomes A and B and remove from it the connected component containing* $\circ$ *to obtain the* internal genome breakpoint *graph* $H(A, B)$*.*

Observation 10. *The internal genome breakpoint graph of single copy genomes is circular.*

Here we introduce *positive* genomes. The internal genome breakpoint graph of positive single copy genomes is a circular positive graph.

Definition 28 (Positive genome)**.** *An adjacency of a genome is* positive*, if it contains one head and one tail extremity. A genome is* positive *if all of its adjacencies are positive.*

By the end of the section we will establish that a 2-break scenario for the internal genome breakpoint graph of two single copy genomes can be interpreted as a 2-break scenario for the internal genome breakpoint graph of two positive single copy genomes.

Lemma 9. *The vertices of a circular graph can be partitioned into two independent sets of equal size.*

Proof. A circular graph is a vertex disjoint union of circles, thus it is enough to prove the statement for a circle C. C has an even number of vertices, take every second one of them. The obtained sub-set of vertices is an independent set of C. Its complement is also an independent set of C, thus establishing the result. □

Definition 29 (Independent sets of a circular graph)**.** *The* independent sets *of a circular graph are the independent sets presented in Lemma 9.*

Observation 11. *The independent sets of a positive circular graph are* $\{1_h, \ldots, n_h\}$ *and* $\{1_t, \ldots, n_t\}$.

Lemma 10. *For a circular graph there exists an isomorphic positive circular graph.*

Proof. Take a circular graph G on $2n$ vertices. Denote its independent sets by V^1 and V^2. $|V^1| = |V^2| = n$, due to Lemma 9. Fix bijections $f^1 : V^1 \to V_h$ and $f^2 : V^2 \to V_t$. Together they define a bijection $f : V^1 \cup V^2 \to V_h \cup V_t$. To every colored edge $(\{u, v\}, col)$ of G assign a colored edge $(\{f(u), f(v)\}, col)$ to obtain a positive circular graph isomorphic to G. □

Theorem 3. *For single copy genomes A and B there exist positive single copy genomes A_p and B_p, such that the internal genome breakpoint graphs $H(A, B)$ and $H(A_p, B_p)$ are isomorphic.*

Proof. $H(A, B)$ is a circular graph, denote its number of vertices by $2n'$. Due to Lemma 10, there exists a circular positive graph H on vertices $\{1_h, \ldots, n'_h\} \cup \{1_t, \ldots, n'_t\}$ isomorphic to $H(A, B)$. For every colored edge $(\{u, v\}, black)$ (respectively $(\{u, v\}, gray)$) of H add an adjacency $\{u, v\}$ to a genome A_p (respectively B_p). The genomes A_p and B_p thus obtained are positive and $H(A_p, B_p) = H$. □

We conclude by establishing that a 2-break scenario for $H(A, B)$ can be interpreted as a 2-break scenario for $H(A_p, B_p)$.

Definition 30 (Image of a 2-break)**.** *Take a function $f : V_1 \to V_2$ between two sets of vertices and a 2-break $\tau = \Big(\big\{\{u, v\}, \{w, s\}\big\} \to \big\{\{u, w\}, \{v, s\}\big\}, col\Big)$ with vertices in V_1. Denote $\Big(\big\{\{f(u), f(v)\}, \{f(w), f(s)\}\big\} \to \big\{\{f(u), f(w)\}, \{f(v), f(s)\}\big\}, col\Big)$, a 2-break with vertices in V_2, by $f(\tau)$.*

Observation 12. *Take two isomorphic graphs G_1 and G_2, their isomorphism f and a 2-break scenario $\rho = (\tau_1, \ldots, \tau_m)$ for G_1. $f(\rho) = (f(\tau_1), \ldots, f(\tau_m))$ is a 2-break scenario for G_2. If ρ is parsimonious, then so is $f(\rho)$.*

2.5.2 Co-tailed genomes

Here we introduce *co-tailed* genomes. A parsimonious 2-break scenario for their breakpoint graph is also a parsimonious 2-break scenario for their internal genome breakpoint graph and vice versa.

Definition 31 (Co-tailed genomes)**.** *Two genomes are co-tailed if their sets of external adjacencies are equal.*

Definition 32 (Circular genome)**.** *A genome is circular if all of its adjacencies are internal.*

Observation 13. *Two circular genomes are necessarily co-tailed.*

The breakpoint graph $G(A, B)$ of co-tailed genomes is a vertex disjoint union of $H(A, B)$ and a terminal graph. This means that $d_{2b}(G(A, B)) = d_{2b}(H(A, B))$, which leads to Lemma 11.

Lemma 11. *Take co-tailed genomes A and B, and parsimonious 2-break scenarios ρ and ρ' for $G(A, B)$ and $H(A, B)$. ρ and ρ' are also parsimonious 2-break scenarios for $H(A, B)$ and $G(A, B)$ respectively.*

2.5.3 Synthesis

By now we are ready to put all the results from this section together. Take two co-tailed single copy genomes A and B and their parsimonious DCJ scenario Δ. Due to Theorem 1, Δ can be interpreted as a parsimonious 2-break scenario $\rho = \rho(\Delta)$ for the breakpoint graph $G(A, B)$. Due to Lemma 11, ρ is also a parsimonious 2-break scenario for the internal genome breakpoint graph $H(A, B)$. Due to Theorem 3, there exist positive genomes A_p and B_p for which $H(A, B)$ and $H(A_p, B_p)$ are isomorphic. Take their isomorphism f. Due to Observation 12, $f(\rho)$ is a parsimonious 2-break scenario for $H(A_p, B_p)$. The latter is a positive circular graph by construction. Due to Lemma 6, there exist permutations σ_1 and σ_2 such that $H(\sigma_1, \sigma_2) = H(A_p, B_p)$. And finally, due to Theorem 2, $f(\rho)$ can be interpreted as a Minimum Length Transposition Decomposition of σ_1 and σ_2. The implicated lemmas and theorems go both ways, thus ensuring that an MLTD of σ_1 and σ_2 can be interpreted as a parsimonious DCJ scenario for A and B. To this end, we obtain a stronger result. Every transposition on σ_1 or σ_2 can be interpreted as a 2-break on $H(\sigma_1, \sigma_2)$, and every 2-break on $H(\sigma_1, \sigma_2) = H(A_p, B_p)$ can be interpreted as a DCJ on A or B. Thus we obtain that every transposition decomposition of σ_1 and σ_2 can be interpreted as a DCJ scenario for A and B of the same length.

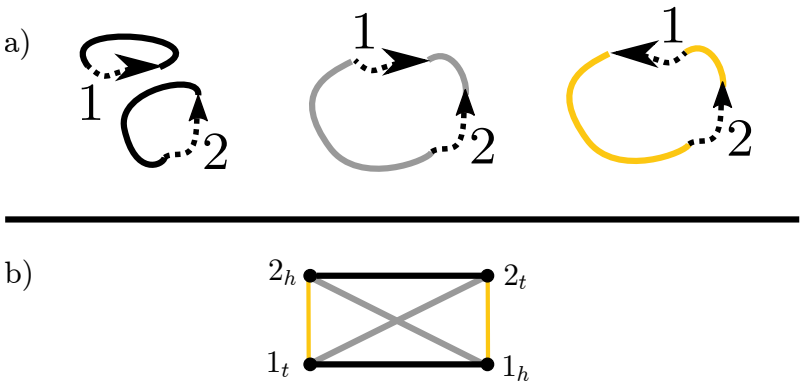


Figure 2.6: Three single copy genomes are depicted in a). The three genomes are circular (see Definition 32), which also means that they are co-tailed (see Definition 31). Their 3-edge-colored internal genome breakpoint graph depicted in b) is not a bipartite graph as was always the case for the internal genome breakpoint graphs of two genomes.

2.6 The Median and Center Problems

2.6.1 The Swap and DCJ Median Problems

The genome breakpoint graph can be easily generalized for 3 genomes. In this case it is a 3-edge-colored multigraph. See Figure 2.6 for an example of the internal genome breakpoint graph of three co-tailed genomes. The internal genome breakpoint $H(A, B, C)$ in the example is not bipartite. This means that the gene extremities no longer can be renamed to obtain three positive genomes with an isomorphic internal genome breakpoint graph. This also means that there are no permutations σ_1, σ_2 and σ_3 with a breakpoint graph $H(\sigma_1, \sigma_2, \sigma_3)$ isomorphic to $H(A, B, C)$. This leads to a situation where the complexity of the Swap Median Permutation problem [82] remains unknown, even though the DCJ Median problem is known to be NP-hard [97].

Fix $n \in \mathbb{N}$ and denote by $\mathbb{G}_n$ a set of all the genomes with genes $\{1, \ldots, n\}$.

Problem 1 (The Swap Median Permutation problem (SMP), open [82])**.**

$$\textit{INPUT}: X \subseteq \mathbb{S}_n, \textit{ and } r \in \mathbb{N}.$$
$$\textit{OUTPUT}: \textit{TRUE, if } \min_{\sigma_M \in \mathbb{S}_n} \left(\sum_{\sigma \in X} d_{Cayley}(\sigma_M, \sigma) \right) \leq r. \textit{ FALSE otherwise.}$$

Observation 14. *Popov [82] uses the term* swap *for what we call* transposition*.*

Problem 2 (The DCJ Median problem for 3 circular genomes, NP-complete [97])**.**

$$\textit{INPUT}: \textit{Three circular genomes } A_1, A_2, A_3, \textit{ and } r \in \mathbb{N}.$$
$$\textit{OUTPUT}: \textit{TRUE, if } \min_{A_M \in \mathbb{G}_n} \left(\sum_{i \in \{1,2,3\}} d_{aDCJ}(A_M, A_i) \right) \leq r. \textit{ FALSE otherwise.}$$

Not only is there no clear reduction from the DCJ Median problem for circular genomes to SMP, but also the reduction from the Breakpoint Graph Decomposition problem, a variant of the Maximum Alternating Edge-disjoint Cycle Decomposition (MAECD) problem, used to establish NP-hardness of a number of other median problems does not seem to apply to SMP.

A graph G is a *balanced bicolored graph* if the black and gray degrees of its vertices are equal to 1 or 2 and it does not contain a monochromatic cycle nor parallel edges. Caprara [28] demonstrated that the Breakpoint Graph Decomposition problem, consisting of finding an MAECD of a balanced bicolored graph, is NP-hard. To prove the NP-hardness of the DCJ Median problem Tannier, Zhen and Sankoff [97] used a reduction from the Breakpoint Graph Decomposition problem similar to the one originally introduced by Caprara [29] for the Inversion Median problem and later used by Feijão and Araujo [44] for the Intermediate Genome Median problem. In that reduction three genomes are associated to a balanced bicolored graph and some properties about the median are established. Two of the genomes and the obtained median are positive, but not the third genome, and there does not seem to be a simple way to modify that reduction so that it could work for SMP.

For now we pose the following question that would further our understanding of the links between sorting genomes by DCJs and sorting permutations by transpositions.

Problem 3 (open problem)**.** *Does a set of positive genomes admit a DCJ median that is positive?*

If the answer to Problem 3 is yes, then the Swap Median Permutation problem is equivalent to the DCJ Median problem for positive genomes, which itself is equivalent to the DCJ Median problem for the co-tailed genomes for which the internal breakpoint graph is bipartite.

2.6.2 The Swap and DCJ Center Problems

The following problems illustrate the opposite situation, where our understanding of sorting with transpositions might inform us on sorting with DCJs.

Problem 4 (The Swap Center Permutation problem (SCP), NP-complete [82])**.**

$$\begin{aligned} &\textit{INPUT}: X \subseteq \mathbb{S}_n, \textit{ and } r \in \mathbb{N}. \\ &\textit{OUTPUT}: \textit{TRUE, if } \min_{\sigma_C \in \mathbb{S}_n} \left(\max_{\sigma \in X} d_{Cayley}(\sigma_C, \sigma) \right) \leq r. \textit{ FALSE otherwise.} \end{aligned}$$

Problem 5 (The DCJ Closest Genome problem, open [34])**.**

$$\begin{aligned} &\textit{INPUT}: \textit{a set } X \textit{ of genomes, and } r \in \mathbb{N}. \\ &\textit{OUTPUT}: \textit{TRUE, if } \min_{A_C \in \mathbb{G}_n} \left(\max_{A \in X} d_{aDCJ}(A_C, A) \right) \leq r. \textit{ FALSE otherwise.} \end{aligned}$$

Problem 6 (open problem)**.** *Does a set of positive genomes admit a DCJ closest genome that is positive?*

If the answer to Problem 6 is yes, then SCP can be reduced to the DCJ Closest Genome problem, which would establish its NP-hardness.

2.6.3 The Rank Median Problem

In this section we briefly discuss a recent line of work on a variant of a median problem that is solvable in polynomial time and relates to the Swap Median Permutation problem.

A square real valued matrix M is *orthogonal* if $M^T M = M M^T = I$, where I is the identity matrix and M^T is the transpose of M. A *permutation* matrix is a binary orthogonal matrix. Such a matrix has a single 1 in each column and each row. A permutation matrix P defines a permutation σ_P, with $\sigma_P(i) = j$ if and only if $P[i][j] = 1$. Analogously a permutation σ defines a permutation matrix. The *rank distance* is defined for square real valued matrices M_1 and M_2, with $d_{rk}(M_1, M_2) = rank(M_1 - M_2)$. The following result relates rank distance for permutation matrices and Cayley distance for the corresponding permutations.

Lemma 12 (Zanetti, Biller, and Meidanis [105])**.** *For permutation matrices P_1 and P_2 we have that $d_{rk}(P_1, P_2) = d_{Cayley}(\sigma_{P_1}, \sigma_{P_2})$.*

Denote the set of all $n \times n$ permutation matrices by $\mathbb{P}^n$.

Problem 7 (The Permutation Rank Median problem for 3 permutation matrices, open [32])**.**

$$\begin{aligned} &\textit{INPUT}: \textit{permutation matrices } P_1, P_2, P_3 \in \mathbb{P}^n, \textit{ and } r \in \mathbb{N}. \\ &\textit{OUTPUT}: \textit{TRUE, if } \min_{P_M \in \mathbb{P}^n} \left(\sum_{i \in \{1,2,3\}} d_{rk}(P_M, P_i) \right) \leq r. \textit{ FALSE otherwise.} \end{aligned}$$

Due to Lemma 12, the Permutation Rank Median problem is equivalent to the Swap Median Permutation problem.

Problem 8 (The Rank Median problem)**.**

$$\textit{INPUT}: \textit{ matrices } M_1, M_2, M_3 \in \mathbb{R}^{n\times n}, \textit{ and } r \in \mathbb{N}.$$
$$\textit{OUTPUT}: \textit{ TRUE, if } \min_{M_M \in \mathbb{R}^{n\times n}} \left(\sum_{i\in\{1,2,3\}} d_{rk}(M_M, M_i) \right) \leq r. \textit{ FALSE otherwise.}$$

Chindelevitch, Zanetti, Pereira and Meidanis [33] established that the Rank Median problem is of polynomial time complexity for three orthogonal matrices. A *Genomic* matrix is defined to be a symmetric permutation matrix. Chindelevitch, La, and Meidanis [32] provided a cubic time algorithm for the Rank Median problem for three genomic matrices and showed that there always exists a rank median that is symmetric and orthogonal, but not necessarily binary.

2.7 Sorting Permutations, Strings and Graphs

2.7.1 Introduction

In the previous sections we have established that the problem of sorting graphs with 2-breaks generalizes those of sorting genomes with DCJs and sorting permutations with transpositions. In this section, without entering too much into the details, we complement the picture by introducing the problems of sorting strings with interchanges and swapping tokens on graphs, together with their cost constrained variants. We also explain how the results on the cost constrained 2-breaks to be presented in this thesis generalize the known results for these seemingly unrelated sorting problems.

2.7.2 The Dutch National Flag Problem (DNF) [21]

Take a string x of n marbles that are either red, white or blue, and a string y with the red marbles going first, followed by the white ones, followed by the blue ones. Any two marbles in x can be interchanged. The Dutch National Flag problem asks for a minimum length interchange scenario transforming x to y.

Bitner [21], represents x with a digraph $D(x)$ on vertices $\{R, B, W\}$ with directed edges $\Big\{(x[i], y[i]) | i \in \{1, \ldots, n\}\Big\}$. See Figure 2.7 a) for an example. He proves that the minimum length of an interchange scenario transforming x to y is equal to $n - c(D(x))$, where $c(D(x))$ is the size of a maximum edge disjoint decomposition of $D(x)$'s edges into directed cycles. Computing $c(D(x))$ is NP-hard and so is the DNF problem.

We define a 2-edge-colored graph $H(x, y)$ on vertices $\{1, \ldots, n\} \cup \{R, W, B\}$ with black edges $\{i, x[i]\}$ and gray edges $\{i, y[i]\}$. $H(x, y)$ is Eulerian and bipartite by construction. See Figure 2.7 b) for an example demonstrating that an interchange on x corresponds to a 2-break on $H(x, y)$ preserving its bipartite structure. The minimum length of an interchange scenario transforming x to y is equal to $d_{2b}(H(x, y))$.

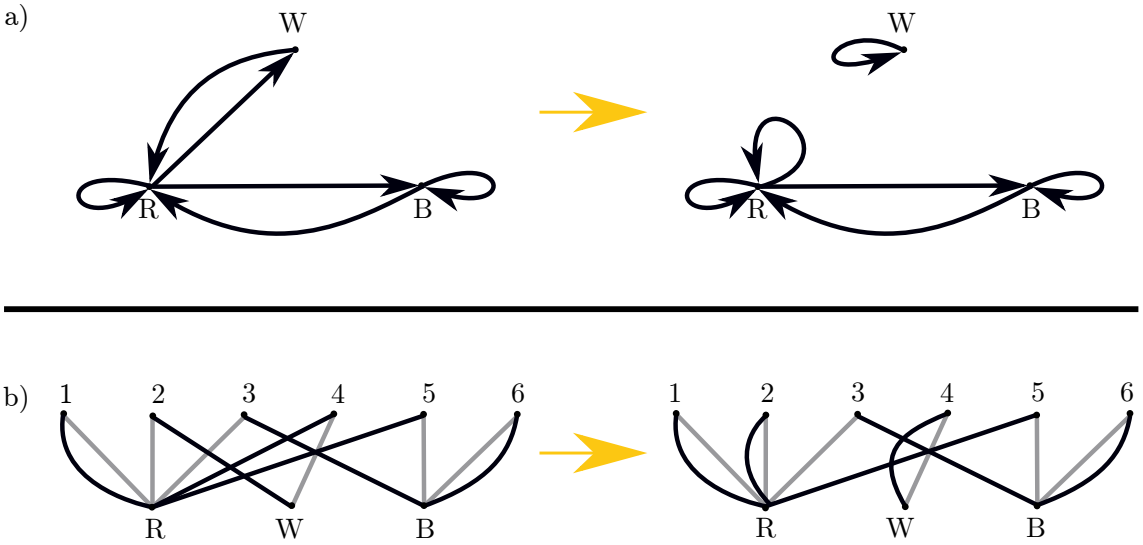


Figure 2.7: First we illustrate how an interchange of two elements in a string can be interpreted as a 2-break on a properly defined graph. Take strings $x = R, W, B, R, R, B$ and $y = R, R, R, W, B, B$. Interchange the second and the fourth elements of x to obtain $x' = R, R, B, W, R, B$. One can observe in a) that a transformation $D(x) \to D(x')$ resembles a 2-break. A transformation $H(x, y) \to H(x', y)$ is a 2-break that preserves the bipartite structure of a graph as shown in b). Two interchanges are enough to transform x into y. We proceed with an illustration of a polynomial reduction from DNF to MIN-COST-TD mentioned in Section 2.7.4. Make the cost of a transposition (ij) to be 0 if $y[i] = y[j]$ and 1 otherwise. Permutation $\sigma_1 = (24)(35)$, satisfies $x[i] = y[\sigma_1(i)]$, and so does $\sigma_2 = (124365)$. The minimum cost of a transposition decomposition for both of these permutations is equal to 2, which is also the minimum number of interchanges required to transform x to y. Transpositions (24) and (35) of cost 1 transform σ_1 to id. Transpositions (16) and (34) of cost 1 transform σ_2 into $\sigma_2' = (231)(56)$, which can be transformed to id with the help of three transpositions of cost 0.

2.7.3 MIN-cost-MLTD [41]

Farnoud and Milenkovic [41] fix an arbitrary cost function on the set of transpositions. They define the cost of a transposition decomposition to be the sum of the costs of its transpositions. The authors propose an $O(n^4)$-time algorithm for the

problem of finding a minimum cost decomposition among the Minimum Length Transposition Decompositions of a permutation (MIN-cost-MLTD) .

This work on the MIN-cost-MLTD problem influenced our research on cost constrained 2-breaks. The results that we will present in Chapter 5 allow us to generalize the work by Farnoud and Milenkovic [41] in a number of ways. First, we allow the cost of a transposition (ij) acting on a permutation σ to depend not only on i and j as in [41], but also on $\sigma^{-1}(i)$ and $\sigma^{-1}(j)$, while keeping the same asymptotic complexity for solving the MIN-cost-MLTD problem. Second, a transposition decomposition in [41] can be seen as sorting a source permutation σ_1 into a target permutation id. Our work, on the other hand, can be interpreted as sorting two permutations σ_1 and σ_2 into some intermediate one, with the transpositions performed on σ_1 and σ_2 possibly having different costs.

2.7.4 MIN-cost-TD [41, 42]

Farnoud and Milenkovic [41] also investigate the problem of finding a minimum cost transposition decomposition of a permutation, the MIN-cost-TD problem. They prove that for an arbitrary cost function MIN-cost-MLTD is a 4-approximation of MIN-cost-TD. The work on the MIN-cost-TD problem [41, 42] combined with our results on relating transposition decompositions and 2-break scenarios could in the future inform the search for a minimum cost 2-break scenario for a graph.

In [41] and [42] it remained open whether the MIN-cost-TD problem is NP-hard. We provide a polynomial time reduction from DNF to MIN-cost-TD. See Figure 2.7 for an example. Take an instance (x, y) of DNF and a permutation $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ for which $x[i] = y[\sigma(i)]$. σ is not necessarily unique, and $\sigma(i)$ fixes a position in y into which an element in position i in x must be moved. An interchange of the elements in the positions i and j in x corresponds to a transposition $(\sigma(i)\sigma(j))$ performed on σ. Make the cost of (ij) to be 0 if $y[i] = y[j]$ and 1 otherwise. It can be shown that for this cost function, the cost of a minimum cost transposition decomposition of σ is equal to the minimum length of an interchange scenario transforming x into y. This establishes that MIN-cost-TD is NP-hard.

2.7.5 The Interchange Rearrangement Problem [4]

Take two strings x and y of length n with equal multisets of elements from a set S. Any two elements in x can be be interchanged. Costs are associated to interchanges and the goal is to transform x to y with a scenario minimizing the sum of the costs of its interchanges. Amir and Levy [4] propose a detailed survey of related Interchange Rearrangement problems. If each string has a unique occurrence of each element, then the problem is equivalent to the MIN-cost-TD problem.

Amir, Hartman, Kapah, Levy, and Porat [3] pose the w-Interchange Distance problem. Here the cost of an interchange of two elements at positions i and j is equal to $|i-j|^\alpha$ with $\alpha \geq 0$. Among other results, the authors propose a linear time algorithm for $\alpha = 1$ and show that the problem is NP-hard for $\alpha = 0$. The Dutch National Flag problem is a particular case with the set S being equal to $\{R, W, B\}$ and $\alpha = 0$.

Kapah, Landau, Levy, and Oz [64] study the Interchange Rearrangement Problem Under the Element-cost Model. Here the cost of an interchange depends on the elements that are being interchanged and not on their positions as in the w-Interchange Distance problem. The authors fix a non-negative function $weight : S \to R_+$ on the elements in the set S. A symmetric function g on the pairs of elements is said to be *general* if for any three elements u, w, s it satisfies that $weight(w) \leq weight(s)$ if and only if $g(u, w) \leq g(u, s)$. The cost of an interchange of u and w is defined to be $g(u, w)$. If each string x and y has a unique occurrence of each element, then a minimum cost interchange scenario for a general function can be found in linear time. The problem becomes NP-hard for general strings, however a 3-approximation algorithm is proposed in [64].

As in Section 2.7.2, we define a bipartite 2-edge-colored graph $H(x, y)$ on vertices $\{1, \ldots, n\} \cup S$ with black edges $\{i, x[i]\}$ and gray edges $\{i, y[i]\}$. $H(x, y)$ is Eulerian by construction. An interchange of elements in x results in a 2-break on $H(x, y)$. Consider the Parsimonious Interchange Rearrangement problem, where we search for a minimum cost interchange scenario among the parsimonious ones.

If each string x and y has a unique occurrence of each element, and the interchange cost depends only on the positions of the elements being interchanged, then this problem is equivalent to MIN-cost-MLTD. Our results in Chapter 5 allow us to solve this problem for a cost function that depends both on the positions and the elements being interchanged. Our algorithm is polynomial for the strings that have unique occurrence of each element. The problem for general strings is NP-hard.

2.7.6 The Token Swapping Problem [100, 23, 101]

Take a connected 1-edge-colored simple graph on n vertices. Place unique tokens on its vertices and assign each token a unique destination vertex. Two tokens lying on adjacent vertices can be swapped. A *token swap sequence* is a sequence of token swaps that move all the tokens to their destination vertices. The Token Swapping problem [100] asks for the minimum length of a token swapping sequence. See Figure 2.8 for an illustration of the problem.

To a token t lying on a vertex u assign a token $\sigma(t)$ whose destination vertex is u. Take two tokens t_1 and t_2. A swap of the tokens lying on their destination vertices transforms σ into $(t_1 t_2)\sigma$, and at the end of the scenario the identity permutation is obtained. This means that the Token Swapping problem is equivalent to

the MIN-COST-TD problem where some transpositions are assigned infinite cost. The complexity of the TOKEN SWAPPING problem remains unknown for trees. Yamanaka et al. [100] observed that there always exists a token swapping sequence of length $O(n^2)$. This might be useful to prove Conjecture 1 from Farnoud, Milenkovic, Puleo, and Su [42] that there always exists an $O(n^2)$ length transposition decomposition of minimum cost.

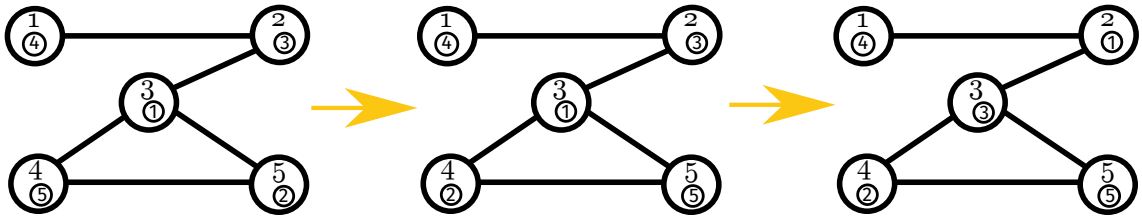


Figure 2.8: On the left we have an instance of the TOKEN SWAPPING problem where the tokens $1, 2, 3, 4, 5$ are placed on the vertices $3, 5, 2, 1, 4$. To this instance we assign a permutation $\sigma_1 = (13254)$. A swap of the tokens lying on the vertices 4 and 5 provide us with a permutation $\sigma_2 = (1324)$, which is equal to $(45)\sigma_1$. A further swap of the tokens lying on the vertices 2 and 3, provide us with a permutation $\sigma_3 = (124)$, which is equal to $(23)\sigma_2$. This instance of the TOKEN SWAPPING can be interpreted as the problem of decomposing (13254) into a product of transpositions from the subset $\{(12), (23), (34), (35), (45)\}$. A possible decomposition is $(45)(23)(12)(23)(34)(23) = (13254)$.

2.7.7 The Colored Token Swapping Problem [23, 101]

Place colored tokens on the colored vertices of a connected 1-edge-colored simple graph. Two tokens lying on adjacent vertices can be swapped. A *colored token swap sequence* is a sequence of token swaps that moves the tokens to the vertices with corresponding colors. The COLORED TOKEN SWAPPING problem asks for the minimum length of a colored token swap sequence. For two colors, the COLORED TOKEN SWAPPING problem can be solved in polynomial time for general graphs and in linear time for trees. For three colors, the problem is NP-hard even for restricted families of graphs [101].

Enumerate the vertices of a graph and take two sequences x and y, where $x[i]$ is the color of the token on the vertex i and $y[i]$ is the color of the vertex i. The COLORED TOKEN SWAPPING problem can be interpreted as the INTERCHANGE REARRANGEMENT problem on x and y, where the interchanges of the elements at the positions corresponding to the adjacent vertices have cost equal to 1 and other interchanges have infinite cost.

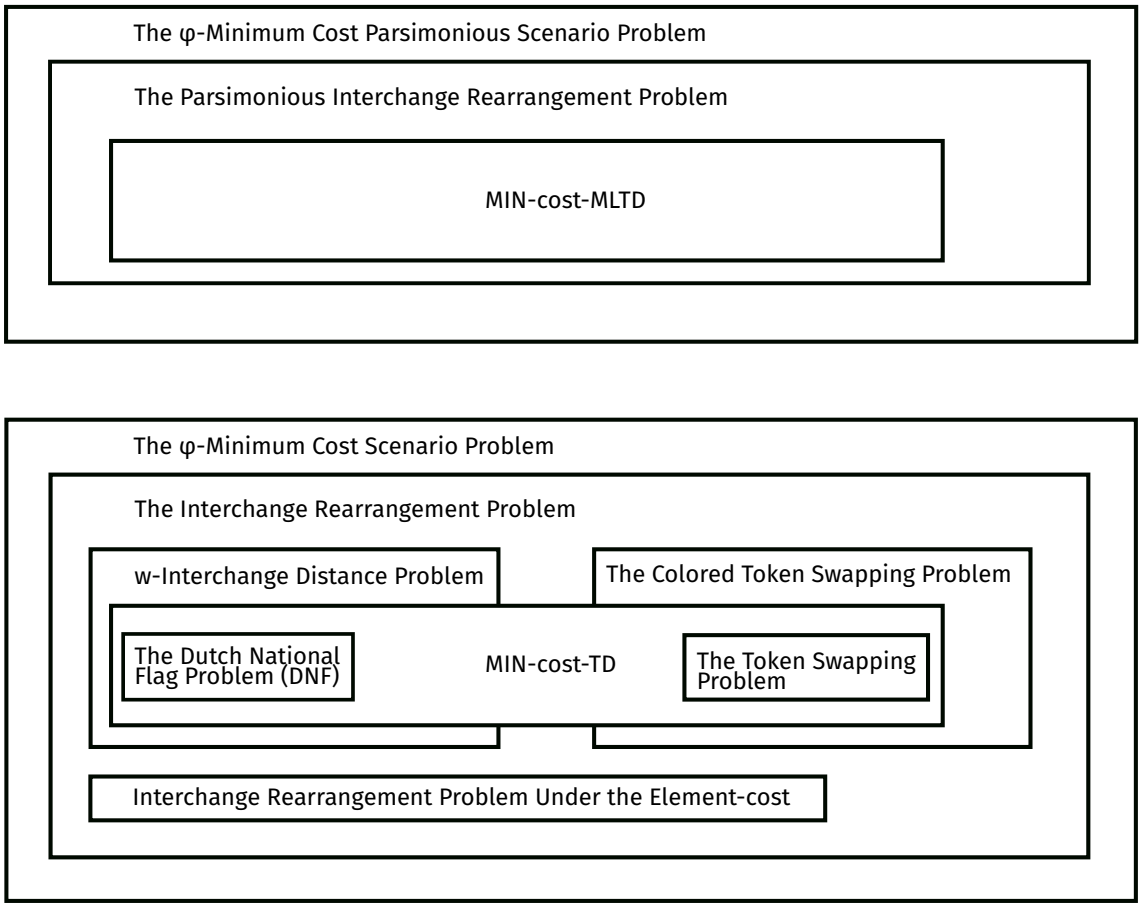


Figure 2.9: In Chapter 4 we introduce the φ-Minimum Cost Parsimonious Scenario and the φ-Minimum Cost Scenario problems for cost constrained 2-breaks that generalize all of the problems presented in Section 2.7. The figure summarizes connections between the sorting problems that we have established in this section. Here a problem A is presented as being included into B if and only if there exists a polynomial time reduction from A to B.

2.8 Conclusion

We have established novel links between the problems of sorting genomes with DCJs, sorting permutations with transpositions and sorting graphs with 2-breaks. See Figure 2.9 for a summary.

These links allow us to interpret a DCJ in a parsimonious sorting scenario for genomes with n genes as a transposition acting on a permutation of n elements, and vice versa. However, this symmetry breaks once a non-parsimonious DCJ scenario is being considered. Feijão and Meidanis [46] showed that a genome with n genes can be interpreted as a permutation of $2n$ elements, and that a DCJ acting on a genome can be interpreted as a pair of transpositions acting on that permutation (see Figure 2.5

for an example). Their work allows us to interpret a non-parsimonious DCJ scenario as a transposition decomposition of a permutation, however the reverse problem of interpreting a transposition decomposition as a DCJ scenario is not addressed. Combination of these two lines of work might lead to a better understanding of the non-parsimonious 2-break and DCJ scenarios.

Such an understanding might allow for an exchange of ideas on sampling and estimation. For example, see a recent work by Irurozki, Calvo, and Lozano [62] on sampling permutations based on their Cayley distance to a given permutation, which might inform us on how to sample genomes based on their DCJ distance to a given genome.

Chapter 3

A Parsimonious 2-break Scenario for a Graph

3.1 Introduction

In this chapter we study in detail the structure of a parsimonious 2-break scenario. We show that a parsimonious 2-break scenario for a graph can be interpreted as a set of parsimonious 2-break scenarios for the circles corresponding to its simple cycles. In addition to that, a parsimonious 2-break scenario for a circle can be partitioned into parsimonious 2-break scenarios for its *sub-circles*, and this property allows for a dynamic programming algorithm exploring the space of the parsimonious 2-break scenarios for a circle. The main results of this chapter are the following:

1. A parsimonious 2-break scenario for a graph can be partitioned into parsimonious 2-break scenarios for the simple cycles in one of its Maximum Alternating Edge-disjoint Cycle Decompositions.

2. A parsimonious 2-break scenario for a simple cycle S can be interpreted as a parsimonious 2-break scenario for a circle arising from one of the *Eulerian orientations* of S.

3. A parsimonious 2-break scenario for a *sub-circle* of a circle C can be partitioned into parsimonious 2-break scenarios for smaller sub-circles of C.

4. There is a bijection between the equivalence classes of the parsimonious 2-break scenarios for a circle and the quadrangulations of a regular polygon. There is also a bijection between the *equivalence classes of the Minimum Length Transposition Decompositions* of a cyclic permutation σ and the equivalence classes of the parsimonious 2-break scenarios for its permutation breakpoint graph $H(\sigma, id)$.

Taken together these results enable us to efficiently search the space of parsimonious 2-break scenarios. We will build on them in Chapter 5 where we search for a minimum cost 2-break scenario among the parsimonious ones. We have published the initial versions of these results in [90, 89].

3.2 A Parsimonious 2-break Scenario for a Graph is a Shuffle of the Parsimonious 2-break Scenarios for the Simple Cycles

In this section we show that a parsimonious 2-break scenario for a graph can be partitioned into parsimonious 2-break scenarios for its simple cycles. We do this with the help of the *trajectory graph* introduced by Shao, Lin, and Moret [87]. The trajectory graph there is defined for a DCJ scenario, and Theorem 2 in [87] establishes that the connected components of the trajectory graph of a parsimonious DCJ scenario are trees, and that these connected components correspond to the cycles of the adjacency graph. We redefine the trajectory graph of a 2-break scenario for a graph G, and show in Theorem 4 that the connected components of the trajectory graph of a parsimonious 2-break scenario for G are trees corresponding to a Maximum Alternating Edge-disjoint Cycle Decomposition of G.

Observation 15. *A shuffle of the 2-break scenarios for the subgraphs in an Eulerian decomposition of a graph is a 2-break scenario for that graph. If the scenarios for the subgraphs are parsimonious, then the obtained scenario for the graph is also parsimonious.*

In this subsection we establish a complementary result. Namely, that for a 2-break scenario ρ on a graph there exists an Eulerian decomposition such that ρ is a shuffle of the 2-break scenarios for its subgraphs.

Definition 33 (Sink/source vertex and edge of a digraph)**.** *A vertex of a 2-edge-colored digraph is a* source*, if its black and gray indegrees are equal to 0. It is a* sink*, if its black and gray outdegrees are equal to 0. An edge of such a digraph is a* source edge*, if it is incident to a source vertex, while it is a* sink edge *if it is incident to a sink vertex. According to this definition an edge can be both a source and a sink at the same time.*

Denote by ρ_l the prefix of ρ of length l and by G_l the graph obtained from G after ρ_l is performed. Construct the *trajectory graph* $\mathcal{D}(G, \rho)$ in the following way. Start with an empty graph. For each colored edge $(\{u, v\}, col)$ of G add two new vertices connected by a directed edge of color col labeled $\{u, v\}$ to obtain $\mathcal{D}(G, \rho_0)$. See Figure 3.1 d) for an example. For the l-th 2-break in ρ

$(\{u, v\}, \{w, s\} \to \{u, w\}, \{v, s\}, col)$, choose any two sink edges of $\mathcal{D}(G, \rho_{l-1})$ of color *col* labeled $\{u, v\}$ and $\{w, s\}$, and merge their sink vertices. Proceed by adding two new edges of color *col* labeled $\{u, w\}$ and $\{v, s\}$ from the merged vertex to the newly added ones, thus obtaining $\mathcal{D}(G, \rho_l)$. Continue until $\mathcal{D}(G, \rho_m)$ is obtained, where m is the length of ρ. G_m is terminal, thus the multisets of the labels of its sink black and gray edges are equal. This means that there exists a bijection between the sink vertices of $\mathcal{D}(G, \rho_m)$ incident to the edges of the same label but of different colors. Choose such a bijection at random and merge the pairs of the sink vertices of $\mathcal{D}(G, \rho_m)$ mapped by this bijection. Denote the obtained graph by $\mathcal{D}(G, \rho)$. See Figure 3.1 for an example of $\mathcal{D}(G, \rho)$.

Shao, Lin, and Moret [87] prove that the connected components of the trajectory graph for a parsimonious scenario are trees, and that they correspond to the cycles of the adjacency graph. The following theorem can be seen as a generalization of this result for the non-parsimonious scenarios.

Theorem 4. *Take a 2-break scenario ρ of length m for a graph G. $\mathcal{D}(G, \rho)$ has a number of connected components k greater than or equal to $e(G) - m$. Further, there exists an Eulerian decomposition $\mathcal{H}$ of G of size k, such that ρ is a shuffle of the 2-break scenarios for its subgraphs. If ρ is parsimonious, then $\mathcal{H}$ is an MAECD of G.*

Proof. Denote by $\{C^1, \ldots, C^k\}$ the connected components of $\mathcal{D}(G, \rho)$. For $i \in [\![1, k]\!]$ and $l \in [\![0, m]\!]$, denote by C^i_l a subgraph of $\mathcal{D}(G, \rho_l)$ consisting of its connected components containing the source vertices of C^i. By construction, there exists a bijection between the sink edges of the graphs in $\{C^1_l, \ldots, C^k_l\}$ and the colored edges of G_l, that maps a sink edge of color *col* labeled $\{u, v\}$ to a colored edge $(\{u, v\}, col)$ in G_l. Denote by H^i_l a subgraph of G_l induced by the colored edges in the image of the sink edges of C^i_l under this bijection. $\{H^1_l, \ldots, H^k_l\}$ is an edge-disjoint decomposition of G_l. See Figure 3.1 for an example of H^i_l.

We prove that H^i_0 is Eulerian by decreasing induction on l. The multisets of the labels of the black and gray sink edges of C^i_m are equal by construction, thus H^i_m is terminal and necessarily Eulerian. Suppose that H^i_l is Eulerian for an $0 < l \leq m$. The two vertices of $\mathcal{D}(G, \rho_{l-1})$ merged during the l-th 2-break in ρ either both belong to C^i_{l-1}, or both are outside C^i_{l-1}. In the first case, H^i_l can be obtained from H^i_{l-1} via a 2-break. A 2-break does not modify the degrees of the vertices and H^i_l is Eulerian due to the induction hypothesis, thus H^i_{l-1} is also Eulerian. In the second case, $H^i_l = H^i_{l-1}$, thus the latter stays Eulerian. This way H^i_0 is Eulerian, denote it by H^i.

$\mathcal{D}(G, \rho_0)$ has $2e(G)$ connected components. The l-th 2-break of ρ merges two vertices of $\mathcal{D}(G, \rho_{l-1})$. This reduces its number of connected components by at most one. $\mathcal{D}(G, \rho)$ is obtained from $\mathcal{D}(G, \rho_m)$ after merging $e(G)$ pairs of vertices. This reduces its number of connected components by at most $e(G)$ and establishes that

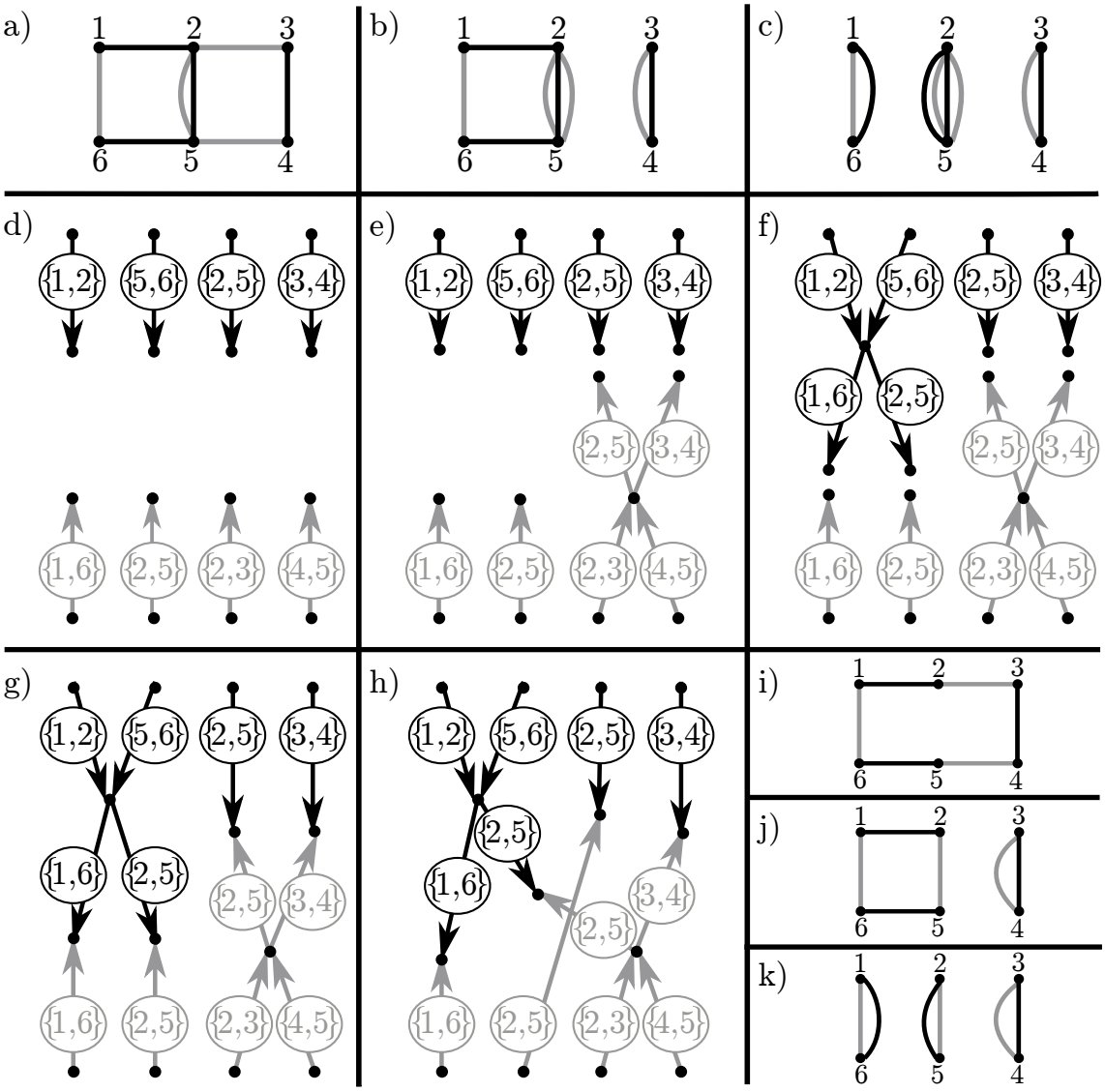


Figure 3.1: An Eulerian 2-edge-colored multigraph G is depicted in a). A parsimonious 2-break scenario $\rho = \Big((\{2,3\},\{4,5\} \to \{2,5\},\{3,4\}, gray),\ (\{1,2\},\{5,6\} \to \{1,6\},\{2,5\}, black)\Big)$ transforms G into a terminal graph G_2 depicted in c). The prefix ρ_1 of ρ of length 1 transforms G into G_1 depicted in b). $\mathcal{D}(G,\rho_0)$, $\mathcal{D}(G,\rho_1)$, and $\mathcal{D}(G,\rho_2)$ are depicted in d), e), and f) respectively. All the edges of $\mathcal{D}(G,\rho_0)$ are both source edges and sink edges. The sink edges of $\mathcal{D}(G,\rho_i)$ correspond to the edges of G_i, while the source edges of $\mathcal{D}(G,\rho_i)$ correspond to the edges of G. Two possibilities for $\mathcal{D}(G,\rho)$ are presented in g) and h), depending on which vertices are chosen to be merged. Denote the larger connected component of $\mathcal{D}(G,\rho)$ in h) by C^1. H_0^1, H_1^1 and H_2^1 are depicted in i), j) and k) respectively.

$k \geq e(G) - m$. If ρ is parsimonious, then $m = e(G) - c(G)$ using Lemma 1 and thus $k \geq c(G)$. Due to the maximality of $c(G)$, G can be partitioned into at most $c(G)$ edge-disjoint Eulerian subgraphs. Due to the result obtained in the previous paragraph, we obtain that $k = c(G)$ and $\mathcal{H} = \{H^1, \ldots, H^k\}$ is an MAECD of G. □

Corollary 4. *A parsimonious 2-break scenario for a graph having two connected components can be partitioned into two sub-sequences that are respectively parsimonious 2-break scenarios for these components.*

3.3 A Parsimonious 2-break Scenario for a Breakpoint Graph

Finding the size of an MAECD of a graph is NP-hard [19]. However we are interested in a particular family of *breakpoint* graphs, corresponding to the genome breakpoint graphs, for which the size of an MAECD can be found in linear time and the space of all the possible MAECDs can be explored efficiently.

This subsection should sound familiar to the reader who is aware of the classical results concerning the parsimonious DCJ scenarios sorting the adjacency graph [25]. Lemma 14 basically reformulates the statement that odd paths and cycles of the adjacency graph are sorted on their own, while Observation 17 restates that an even length path can be recombined with another even length path or sorted on its own during a parsimonious DCJ scenario.

Definition 34 (Single/multiple vertex)**.** *A vertex is* single *if its black and gray degrees are equal to 1 and it is* multiple *if these degrees are larger.*

Definition 35 (Breakpoint graph)**.** *A graph is a* breakpoint graph *if it has at most one multiple vertex. If such a vertex exists, denote it by* $\circ$*.*

Observation 16. *The genome breakpoint graph (see Definition 15) is a breakpoint graph.*

See Figure 3.2 for an example of a breakpoint graph.

Definition 36 (Simple and circle subgraphs)**.** *A subgraph is a* simple subgraph *if it is a simple cycle and a* circle subgraph *if it a circle (see Definition 9). A subgraph is a* simple non-circle subgraph *if it is a simple cycle that is not a circle.*

Lemma 13. *A circle subgraph of a breakpoint graph belongs to its every MAECD.*

Proof. Take a circle subgraph C of a breakpoint graph G and its MAECD $\mathcal{H}$. If C does not include the multiple vertex $\circ$, then it is a connected component of G that

is a simple cycle and thus appears in all the MAECDs of G. Suppose that C does include the multiple vertex $\circ$. Take the black edge of C incident to $\circ$. This colored edge belongs to some Eulerian simple cycle H in $\mathcal{H}$. Take an Eulerian tour Δ of H starting from $\circ$ with this colored edge. All the vertices of G except $\circ$ are single, this means that Δ finishes at $\circ$ with a gray edge. H being a simple cycle means that $H = C$, and thus $C \in \mathcal{H}$. □

We define AA and BB paths of a breakpoint graph as in Definition 16.

Definition 37 (AA/BB path of a breakpoint graph)**.** *Take a connected non Eulerian sub-graph H of G in which the black and gray degrees of every vertex different from $\circ$ are equal to 1. If the black and gray degrees of $\circ$ are respectively equal to 2 and 0, then H is an AA* path *of G. If these degrees are respectively equal to 0 and 2, then H is a BB* path *of G.*

Observation 17. *A simple non-circle subgraph of G is a union of an AA and a BB path.*

See Figure 3.2 for an example. Denote by $\mathcal{B}(G)$ a complete bipartite graph having the AA and the BB paths of G as vertices. Due to Lemma 13 and Observation 17 we obtain the following result.

Lemma 14. *A Maximum Alternating Edge-disjoint Cycle Decomposition of a breakpoint graph G can be identified with a perfect matching of $\mathcal{B}(G)$ plus the set of the circle subgraphs of G.*

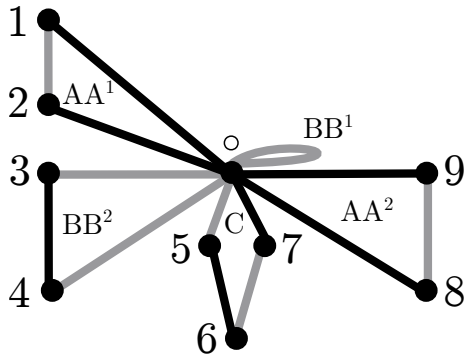


Figure 3.2: An example of a breakpoint graph is depicted that contains a single circle subgraph denoted by C. It also contains two AA paths AA^1 and AA^2, and two BB paths BB^1 and BB^2. This breakpoint graph has only two possible MAECDs that are $\{C, AA^1 \cup BB^1, AA^2 \cup BB^2\}$ and $\{C, AA^1 \cup BB^2, AA^2 \cup BB^1\}$.

3.4 A 2-break Scenario for a Simple Cycle is a 2-break Scenario for a Circle

3.4.1 Introduction

In Section 3.2 we have shown that a parsimonious 2-break scenario for a graph can be partitioned into parsimonious 2-break scenarios for the subgraphs in its Maximum Alternating Edge-disjoint Cycle Decomposition. Due to maximality, these subgraphs themselves have a MAECD of size equal to 1. A graph satisfying this property is a simple cycle, while a circle is a simple cycle with the black and gray degrees of every vertex equal to 1.

In this section we study simple cycles in detail and establish that a 2-break scenario of a simple cycle S corresponds to a 2-break scenario for a circle obtained from an Eulerian orientation of S. This relation will allow us to explore the space of the 2-break scenarios for a simple cycle by exploring the space of the 2-break scenarios for its circles.

The work presented in this section generalizes Proposition 6 from Braga and Stoye [25]. There it is stated that there are two options for a parsimonious DCJ scenario Δ sorting the adjacency graph consisting of an AA and a BB path. Namely, either Δ sorts the paths separately or corresponds to a DCJ scenario for an adjacency graph obtained by joining the paths into a cycle in one of the two possible ways. See Figure 3.3 for an illustration that these two ways of joining the paths into a cycle can be interpreted as two ways of splitting a simple cycle into a circle.

3.4.2 The Maximum Degree of a Simple Cycle is 2

A simple cycle might have some vertices with black and gray degrees higher than 1. Here we establish that these degrees cannot be higher than 2.

Lemma 15. *The black and gray degrees of a vertex of a simple cycle are at most equal to 2.*

Proof. Take a simple cycle S and a vertex v of S. S is Eulerian by definition, thus the black and gray degrees of v are equal, denote them by d. Take an alternating Eulerian tour Δ of S starting at v, and index the colored edges incident to v based on their order of appearance in Δ. If a loop $(\{v, v\}, col)$ is present in S, then assign it two indices as in Figure 3.4. This way we obtain a sequence $(col_1, \ldots, col_{2d})$, where col_i is the color of the edge indexed with i.

If $col_{2i-1} \neq col_{2i}$ for $0 < i < d + 1$, then a contiguous subsequence of Δ starting from v with a colored edge indexed $2i - 1$ and finishing at v with a colored edge

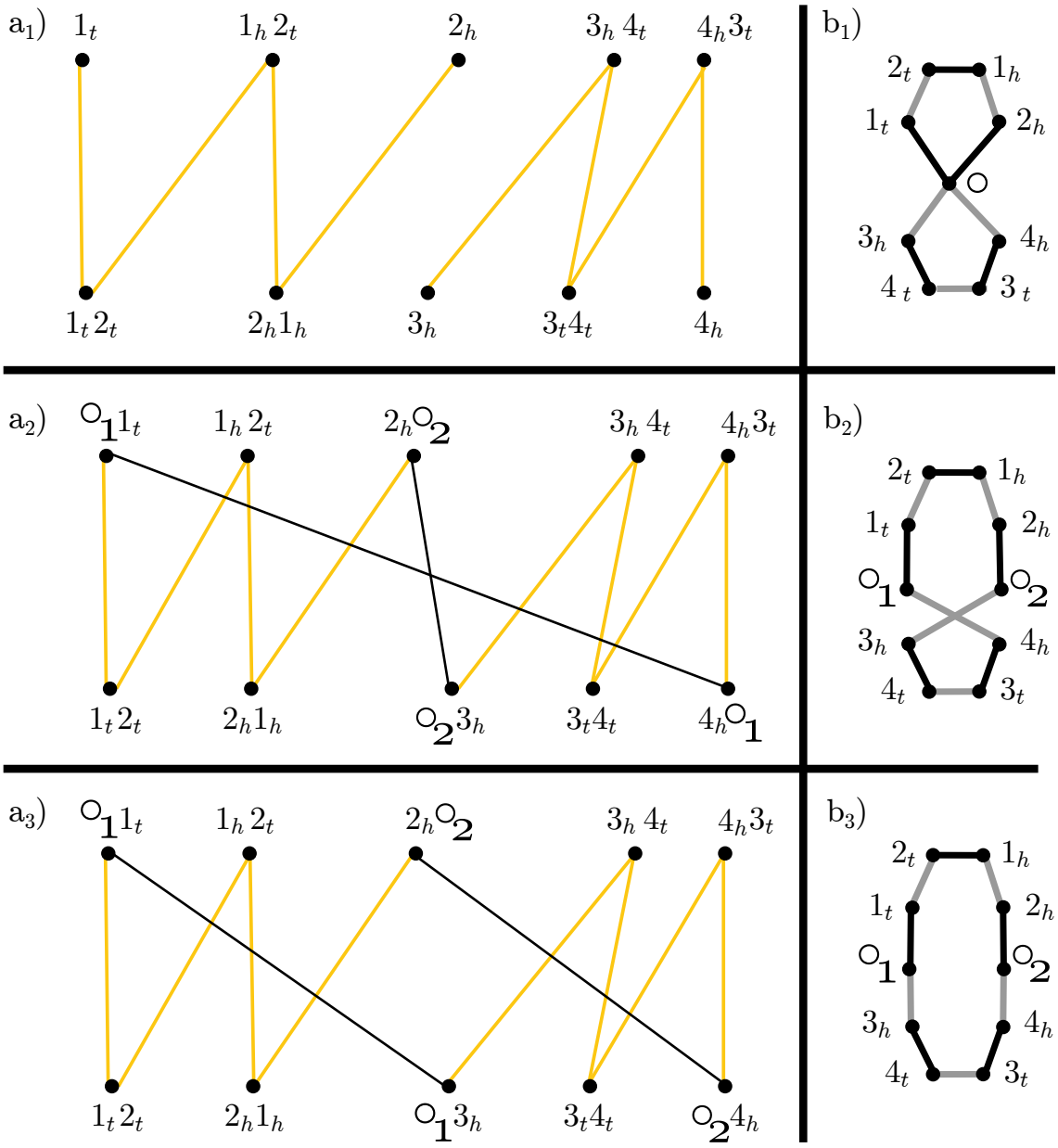


Figure 3.3: In a_1) the adjacency graph $AG(A, B)$ is depicted consisting of an AA and a BB path. The two ways to join these paths into a cycle are depicted in a_2) and a_3). In b_1) a subgraph of the breakpoint graph $G(A, B)$ is depicted. It is a simple cycle obtained by eliminating the colored edges $(\{\circ, \circ\}, black)$ and $(\{\circ, \circ\}, gray)$. b_2) and b_3) depict the circles obtained once its vertex $\circ$ is split into two vertices $\circ_1$ and $\circ_2$. These circles correspond to the cycles depicted in a_2) and a_3). Proposition 6 from Braga and Stoye [25] can be interpreted as stating that a parsimonious 2-break scenario for the simple cycle in b_1) corresponds to a parsimonious 2-break scenario for one of the two circles. In this section establish an analogous result for all the simple cycles and the 2-break scenarios that are not necessarily parsimonious.

indexed $2i$ composes an alternating cycle of S. By definition, the only alternating cycle of S is itself, thus $2i - 1 = 1$, $2i = 2d$ and $d = 1$.

Half of the colors in $(col_1, \ldots, col_{2d})$ are black and the other half are gray. Suppose that $d > 1$, then we have already proven that $col_{2i-1} = col_{2i}$ for all $0 < i < d + 1$. This means that there exists $0 < i < d - 1$ with $col_{2i} \neq col_{2i+1}$. We know that $col_{2i-1} = col_{2i}$ and $col_{2i+1} = col_{2i+2}$. This means that the contiguous subsequence of Δ starting from v with a colored edge indexed $2i - 1$ and finishing at v with a colored edge indexed $2i + 2$ composes an alternating cycle. By definition, the only alternating cycle of S is itself, thus $2i - 1 = 1$, $2i + 2 = 2d$ and $d = 2$. □

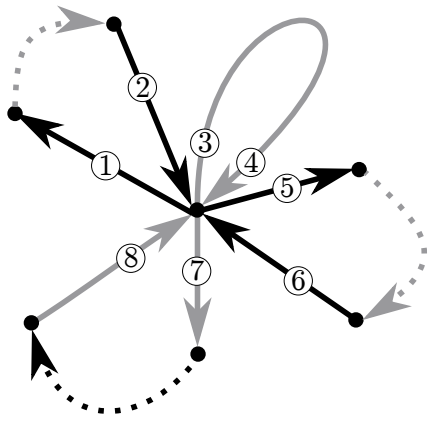


Figure 3.4: This is an illustration for Lemma 15. It depicts an Eulerian tour Δ starting at the central vertex with black and gray degrees equal to 4. Dashed arrows here represent the portions of the tour that do not visit v. For this example the sequence of the colors used in the proof is $\{black, black, gray, gray, black, black, gray, gray\}$. In this case $col_{2i-1} = col_{2i}$, for all i, however the contiguous subsequence of Δ starting with the colored edge indexed by 1 and finishing with the colored edge indexed by 4 comprises an alternating cycle.

3.4.3 Splitting a Double Vertex of a Simple Cycle

Definition 38 (Single/double vertex)**.** *A vertex of a graph is* double *if its black and gray degrees are equal to 2, it is* single *if these degrees are equal to 1.*

In the previous section we have established that all the vertices of a simple cycle are either single or double. In this section we replace a double vertex v with two single vertices v^1 and v^2 in a simple cycle S and its 2-break scenario ρ to obtain a 2-break scenario $\hat{\rho}$ for a simple cycle $\hat{S}$ with one less double vertex than S. Once repeated for every double vertex, this process will leave us with a 2-break scenario for a *circle of* S.

Take a double vertex v of a simple cycle S on vertices V. Replace v in V with vertices v^1 and v^2 to obtain a set of vertices $\hat{V}$. Denote by M a function that transforms a graph on vertices V into a graph on vertices $\hat{V}$ by merging the vertices v^1 and v^2 into v. For a 2-break $\hat{\tau}$ on vertices $\hat{V}$, denote by $M(\hat{\tau})$ a 2-break on vertices V obtained by replacing the occurrences of v^1 and v^2 with v. For a sequence of 2-breaks $\hat{\rho} = (\hat{\tau_1}, \ldots, \hat{\tau_m})$ on vertices $\hat{V}$, denote by $M(\hat{\rho}) = (M(\hat{\tau_1}), \ldots, M(\hat{\tau_m}))$ a sequence of 2-breaks on vertices V.

See Figure 3.5 for all the possible cases of S and $\hat{S}$ such that $M(\hat{S}) = S$. After inspecting the figure it should be clear that the following observation holds.

Observation 18. *There exists a simple cycle $\hat{S}$ such that $M(\hat{S}) = S$ and its vertices v^1 and v^2 are single.*

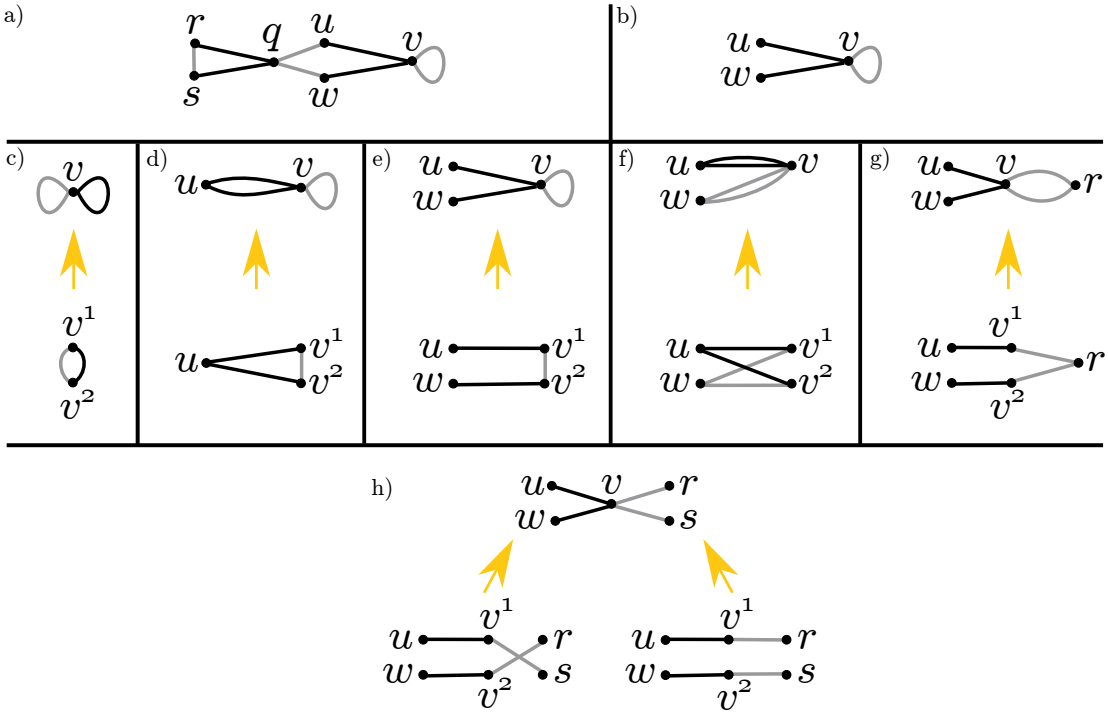


Figure 3.5: An example of a simple cycle S is depicted in a). A subgraph of S induced by the colored edges incident to its double vertex v is presented in b). In total, there are 6 possibilities for such an induced subgraph, modulo an inversion of the colors of the edges. These induced subgraphs are depicted at the top parts of the figures c)-h). At the bottom of the figures c)-h) are depicted the possibilities, modulo an exchange of the names of the vertices v^1 and v^2, of such induced subgraphs of $\hat{S}$, that are transformed into S once their vertices v^1 and v^2 are merged into v.

Lemma 16. *Take a simple cycle S with a double vertex v and a simple cycle $\hat{S}$ satisfying $M(\hat{S}) = S$. For a 2-break τ transforming S into some S', there exists a 2-break $\hat{\tau}$ transforming $\hat{S}$ into some $\hat{S}'$, satisfying $M(\hat{S}') = S'$ and $M(\hat{\tau}) = \tau$.*

Proof. Take a 2-break $\tau = \Big(\big\{\{u,r\},\{w,s\}\big\} \to \big\{\{u,w\},\{r,s\}\big\}, col\Big)$ transforming S into S'. As $M(\hat{S}) = S$, $\hat{S}$ contains two separate colored edges $(\{\hat{u},\hat{r}\}, col)$ and $(\{\hat{w},\hat{s}\}, col)$ that are transformed into colored edges $(\{u,r\}, col)$ and $(\{w,s\}, col)$ once the vertices v^1 and v^2 are replaced with v. A 2-break $\hat{\tau} = \Big(\big\{\{\hat{u},\hat{r}\},\{\hat{w},\hat{s}\}\big\} \to \big\{\{\hat{u},\hat{w}\},\{\hat{r},\hat{s}\}\big\}, col\Big)$ satisfies the statement. See Figure 3.7 a) and b) for an example. □

Lemma 17. *Take a simple cycle S and its 2-break scenario ρ. There exists a simple cycle $\hat{S}$ and its 2-break scenario $\hat{\rho}$ satisfying $M(\hat{S}) = S$ and $M(\hat{\rho}) = \rho$.*

Proof. See Figure 3.7 for an example. Denote by $\overline{S}$ the terminal graph that is obtained from S once ρ is performed. Due to Observation 18, there exists a simple cycle $\hat{S}_1$ on vertices $\hat{V}$ satisfying $M(\hat{S}_1) = S$. Due to Lemma 16, there exists a sequence of 2-breaks $\hat{\rho}_1$ transforming $\hat{S}_1$ into $\overline{\hat{S}_1}$ satisfying $M\Big(\overline{\hat{S}_1}\Big) = \overline{S}$. If $\overline{\hat{S}_1}$ is terminal, then we are done as $\hat{\rho}_1$ is a 2-break scenario for $\hat{S}_1$.

Suppose that $\overline{\hat{S}_1}$ is not terminal. If S has a black loop incident to v, then take $\hat{S}_2 = \hat{S}_1$. Otherwise $\hat{S}_1$ has 2 black edges incident to v^1 and v^2, denote them by $(\{v_1,u\}, black)$ and $(\{v_2,w\}, black)$, with u and w possibly equal. Replace them with $(\{v_2,u\}, black)$ and $(\{v_1,w\}, black)$ to obtain a simple cycle $\hat{S}_2$. By construction $M(\hat{S}_2) = S$. Transform $\hat{\rho}_1$ by replacing all the occurrences of v^1 in the black 2-breaks of $\hat{\rho}_1$ with v^2 and vice versa. Denote the newly obtained sequence of 2-breaks by $\hat{\rho}_2$. Denote the black edges in $\overline{\hat{S}_1}$ incident to v by $(\{v_1,u\}, black)$ and $(\{v_2,w\}, black)$. $\overline{\hat{S}_2}$ can be obtained from $\overline{\hat{S}_1}$ by replacing these colored edges with $(\{v_2,u\}, black)$ and $(\{v_1,w\}, black)$. All the possible cases of $\overline{S}$ and $\overline{\hat{S}_1}$ are depicted in Figure 3.6. After inspecting the figure it should be clear that $\overline{\hat{S}_2}$ is a terminal graph. Thus $\hat{\rho}_2$ is a 2-break scenario for $\hat{S}_2$. □

3.4.4 A 2-break Scenario for a Circle of a Simple Cycle

In this subsection we introduce the notion of a *circle* of a simple cycle S and show that a 2-break scenario for S corresponds to a 2-break scenario for one of its circles.

Take a simple cycle S on vertices V. All the vertices of S are either single or double due to Lemma 15. For every double vertex v in V, replace it with v^1 and v^2 to obtain a set of vertices $\hat{V}$. Denote by M a function that transforms a graph on vertices V into a graph on vertices $\hat{V}$ by merging the pairs of vertices corresponding to the double vertices of S. For a 2-break $\hat{\tau}$ on vertices $\hat{V}$, denote by $M(\hat{\rho})$ a 2-break on vertices V obtained by replacing the occurrences of v^1 and v^2 with v for every double vertex of S. For a sequence of 2-breaks $\hat{\rho} = (\hat{\tau_1}, \ldots, \hat{\tau_m})$ on vertices $\hat{V}$, denote by $M(\hat{\rho}) = (M(\hat{\tau_1}), \ldots, M(\hat{\tau_m}))$ a sequence of 2-breaks on vertices V.

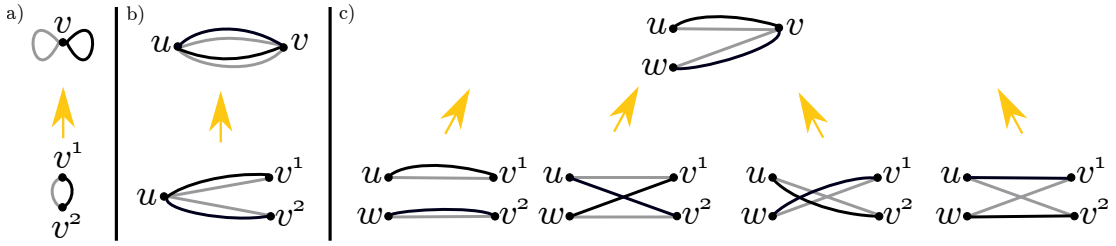


Figure 3.6: Take a simple cycle S and a 2-break scenario ρ transforming S into a terminal graph $\overline{S}$. There are three possibilities for a subgraph of $\overline{S}$ induced by the colored edges incident to a double vertex v of S. These are depicted at the top. At the bottom are the possibilities of such induced subgraphs of $\overline{\hat{S}}$, a graph that is transformed into $\overline{S}$ once its vertices v^1 and v^2 are merged into v. If $\overline{\hat{S}}$ is not already terminal, then in order to obtain a terminal graph it is enough to swap the endpoints of its black edges incident to v^1 and v^2. This means that if $\overline{\hat{S}_1}$ in the proof of Lemma 17 is not terminal, then $\overline{\hat{S}_2}$, obtained by swapping its black edge, is terminal.

Definition 39 (Circle of a simple cycle)**.** *A circle C (see Definition 9) on vertices $\hat{V}$ is a* circle of a simple cycle *S if $M(C) = S$.*

By applying Lemma 17 for every double vertex of a simple cycle we establish the following theorem.

Theorem 5. *Take a 2-break scenario ρ for a simple cycle S. There exists a circle C of S and its 2-break scenario $\hat{\rho}$ such that $M(\hat{\rho}) = \rho$.*

3.4.5 A Link Between the Eulerian Orientations of a Simple Cycle and its Circles

The motivation behind this subsection is to come up with a way to enumerate the circles of a simple cycle S. To this end, we introduce a notion of an *Eulerian orientation of an undirected graph* and show that an Eulerian orientation $\overrightarrow{S}$ of a simple cycle S has a unique directed alternating Eulerian tour starting at a given vertex. We explain how to transform such a tour into a circle of S, and prove that all the circles of S (up to renaming of the double vertices) can be obtained this way.

Definition 40 (Eulerian orientation of a graph)**.** *An* Eulerian orientation *of an undirected 2-edge-colored graph G is an assignment of a direction to each colored edge of G satisfying that the black indegree (respectively outdegree) is equal to the gray outdegree (respectively indegree) for every vertex v.*

See Figure 3.8 for an example of an Eulerian orientation of a simple cycle and a circle.

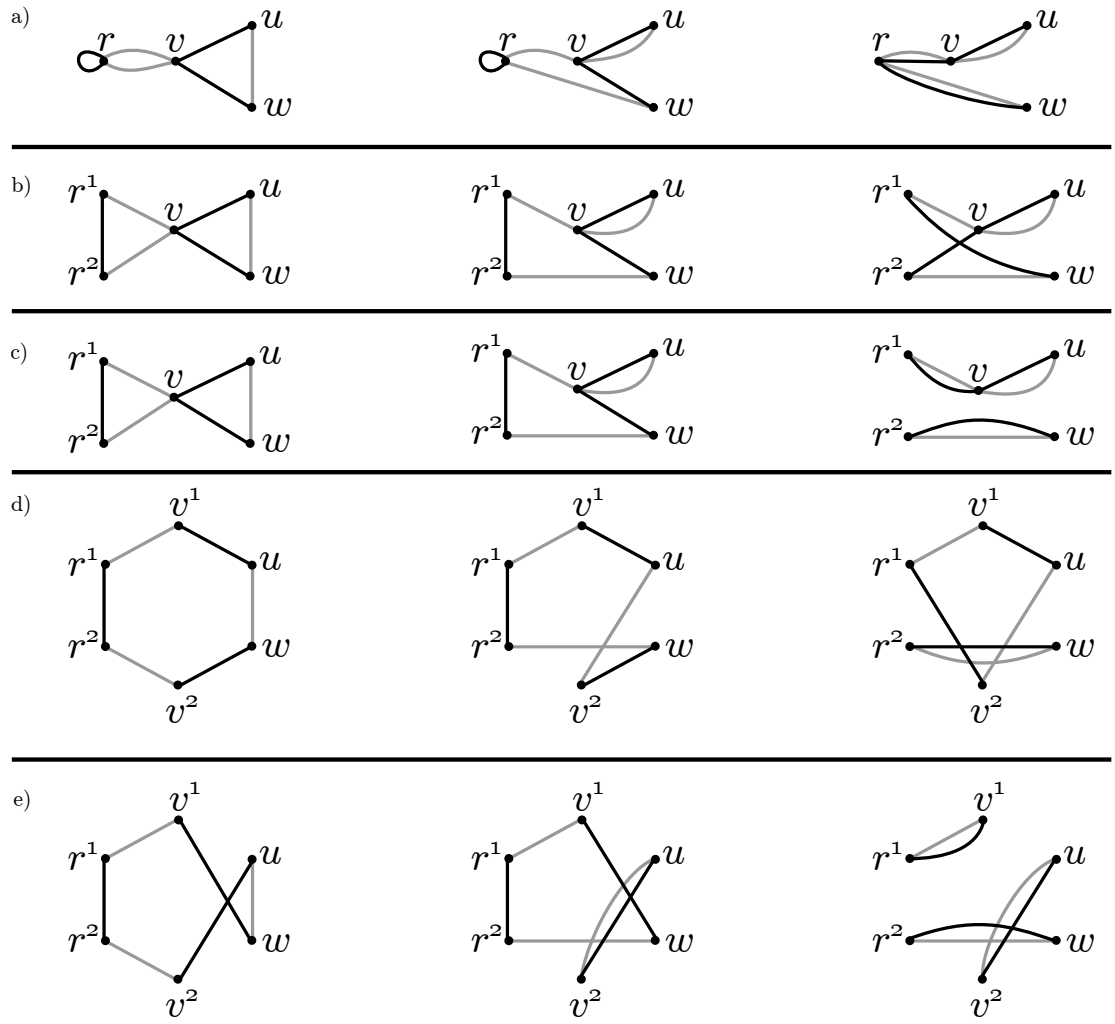


Figure 3.7: A simple cycle S with double vertices r and v is depicted in a) together with a 2-break scenario ρ of length 2 transforming S into a terminal graph $\overline{S}$. A graph $\hat{S}_1$, that becomes S once the vertices r^1 and r^2 are merged into r, is depicted in b) together with a sequence of 2-breaks $\hat{\rho}_1$ satisfying $M(\hat{\rho}_1) = \rho$. For the first 2-break τ_1 in ρ $(\{r, v\}, \{u, w\} \to \{r, w\}, \{u, v\}, gray)$ we have two options for $\hat{\tau}_1$ satisfying $M(\hat{\tau}_1) = \tau_1$: $(\{r^1, v\}, \{u, w\} \to \{r^1, w\}, \{u, v\}, gray)$ and $(\{r^2, v\}, \{u, w\} \to \{r^2, w\}, \{u, v\}, gray)$. We include the second one into $\hat{\rho}_1$. For the second 2-break τ_2 $(\{r, r\}, \{v, w\} \to \{r, v\}, \{r, w\}, black)$ of ρ we also have two options for $\hat{\tau}_2$ satisfying $M(\hat{\tau}_2) = \tau_2$: $(\{r^1, r^2\}, \{v, w\} \to \{r^1, w\}, \{r^2, v\}, black)$ and $(\{r^1, r^2\}, \{v, w\} \to \{r^1, v\}, \{r^2, w\}, black)$. We include the second one into $\hat{\rho}_1$. $\hat{\rho}_1$ transforms $\hat{S}_1$ into a graph $\overline{\hat{S}_1}$ that is not terminal, indicating that some of the choices should be modified. S contains a black loop incident to r, thus we choose $\hat{S}_2 = \hat{S}_1$. In the black 2-breaks of $\hat{\rho}_1$ we swap the vertices v^1 and v^2 to obtain a 2-break scenario $\hat{\rho}_2$ for $\hat{S}_2$ depicted in c). Denote $\hat{S}_2$ by S' and $\hat{\rho}_2$ by ρ'. We proceed the same way in d) with a circle $\hat{S'}_1$, that becomes S' once the vertices v^1 and v^2 are merged into v. Once again we start by choosing $\hat{\rho'}_1$, that does not lead to a terminal graph. To deal with this, we have to swap the black edges of $\hat{S'}_1$ incident to v^1 and v^2 and swap v^1 and v^2 in the black 2-breaks of $\hat{\rho'}_1$ to obtain $\hat{S'}_2$ and $\hat{\rho'}_2$ as depicted in e).

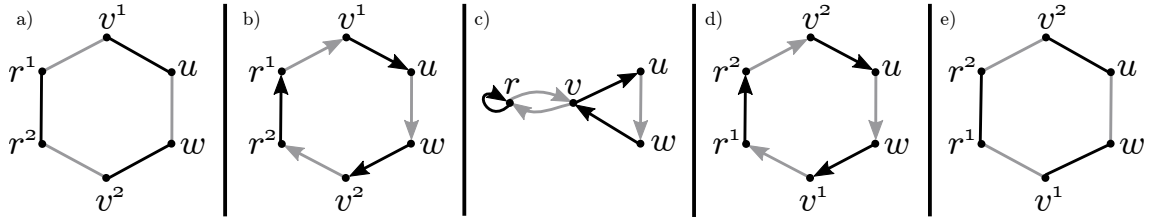


Figure 3.8: A circle C of a simple cycle S from Figure 3.7 is presented in a). In b) one of its two Eulerian orientations is depicted. Merge its vertices v^1 and v^2 into v and its vertices r^1 and r^2 into r to obtain an Eulerian orientation $\overrightarrow{S}$ in c). The u-tour of $\overrightarrow{S}$ is the directed alternating Eulerian tour of $\overrightarrow{S}$ starting from u. Its vertex sequence is (u, w, v, r, r, v, u). The colored directed edges of the u-tour of $\overrightarrow{S}$ incident to the second occurrence of v in the vertex sequence of the tour are $((v, u), black)$ and $((r, v), gray)$. Rename v to v^2 to obtain $((v^2, u), black)$ and $((r, v^2), gray)$. Rename v to v^1 in the rest of the colored directed edges of the tour to obtain $((w, v^1), black)$ and $((v^1, r), gray)$. Proceed by renaming r to obtain an Eulerian orientation of a circle of S depicted in d), that provides the u-circle (see Definition 41) of $\overrightarrow{S}$ depicted in e) once its orientation is forgotten. The circles depicted in a) and e) are equivalent, as it suffice to rename the vertices v^1 and v^2, and r^1 and r^2 in order to transform one circle into another.

Lemma 18. *Take an Eulerian orientation $\overrightarrow{S}$ of a simple cycle S and its double vertex u. The black in and outdegrees of u in $\overrightarrow{S}$ are equal to 1, and so are its gray in and outdegrees.*

Proof. Either the black or the gray outdegree of u must be non-zero and without loss of generality we can suppose that this is the case for the black outdegree. Take a maximum length directed alternating path Δ starting at u with a black edge. Due to the definition of an Eulerian orientation, this path finishes at u with a gray edge. Δ is an Eulerian tour of S due to the simplicity of S. u is a double vertex, thus Δ must have two incoming colored edges to u, and the first one of them must be black, while the second must be gray. This ensures that the black in and outdegrees of u are equal to 1, and so are its gray in and outdegrees. □

Corollary 5 (u-tour of $\overrightarrow{S}$)**.** *For a single vertex u of a simple cycle S there exists a unique directed alternating Eulerian tour of $\overrightarrow{S}$ starting from u. For a double vertex u of S, there exists a unique directed alternating Eulerian tour of $\overrightarrow{S}$ starting with a black edge from u. We call this tour the* u-tour of $\overrightarrow{S}$.

Definition 41 (the u-circle of $\overrightarrow{S}$ and a u-circle of S)**.** *For every double vertex v of S perform the following. Rename v with v^2 in the colored directed edges of the u-tour of $\overrightarrow{S}$ incident to the second occurrence of v in the vertex sequence of that tour. Rename the rest of the occurrences of v in the colored directed edges of the*

u-tour of $\overrightarrow{S}$ *with* v^1*, thus obtaining a set of the colored directed edges of an Eulerian orientation* $\overrightarrow{C}$ *of a circle* C *of* S*.* C *is the* u-circle *of* $\overrightarrow{S}$ *and a* u-circle of S.

See Figure 3.8 for an example of the u-circle of $\overrightarrow{S}$. We define two circles of S to be *equivalent* if they are equal up to switching the names of the pairs of vertices corresponding to the double vertices of S.

Definition 42 (Equivalent circles of a simple graph)**.** *For a simple graph* S*, a graph isomorphism* $g : \hat{V} \to \hat{V}$ *is* S-preserving *if it maps each single vertex of* S *to itself, and for each double vertex* v *of* S *one has either* $g(v^1) = v^1$ *and* $g(v^2) = v^2$*, or* $g(v^1) = v^2$ *and* $g(v^2) = v^1$*. Two circles of* S *are* equivalent *if there exists an* S-preserving *isomorphism between them.*

Theorem 6. *Take a vertex* u *of a simple cycle* S*. For a circle* C *of* S *there exists a* u*-circle of* S *equivalent to* C*.*

Proof. Choose an Eulerian orientation $\overrightarrow{C}$ of a circle C. For every double vertex v in S, merge the vertices v^1 and v^2 in $\overrightarrow{C}$ to obtain an Eulerian orientation $\overrightarrow{S}$ of S. From Figure 3.8 it should be clear that the u-circle of $\overrightarrow{S}$ and C are equivalent. □

3.5 A Parsimonious 2-break Scenario for a Circle

3.5.1 Introduction

In Section 3.2 we have concisely represented a 2-break scenario with the help of the trajectory graph. Shao, Lin, and Moret [87] demonstrated that the trajectory graph $\mathcal{D}(C, \rho)$ of a parsimonious 2-break scenario ρ for a circle C is a tree. Here we present the *scenario graph* $\mathcal{S}(C, \rho)$ that can be interpreted as a particular planar embedding of the trajectory graph, and show that it is a quadrangulation of a regular polygon. We use this enriched graphical tool to partition ρ into parsimonious 2-break scenarios for the sub-circles of C in Theorem 8. In Chapter 5 this partition will lead to a dynamic programming algorithm for finding a minimum cost scenario among the parsimonious ones.

We also establish a bijection between the equivalence classes of the parsimonious 2-break scenarios for a circle and the *matched quadrangulations* of a regular polygon. This work relates to that presented in Section 3 of Dulucq and Penaud [38] and Section IV of Farnoud and Milenkovic [41]. These papers discuss a bijection between the *equivalence classes* of the Minimum Length Transposition Decompositions of a cyclic permutation and a family of the *spanning planar trees on a circle* is established. In Subsection 3.6.4 we combine these results to establish a bijection between the equivalence classes of the MLTDs of a cyclic permutation σ and the

equivalence classes of the parsimonious black-2-break scenarios for the permutation breakpoint graph $H(\sigma, id)$.

3.5.2 The Scenario Graph of a Parsimonious 2-break Scenario for a Circle is Planar

In this subsection we introduce the *scenario graph* of a parsimonious 2-break scenario for a circle and prove that it is planar.

Definition 43 (Scenario graph)**.** *For a circle C and its parsimonious 2-break scenario ρ, define a 1-edge-colored graph $\mathcal{S}(C, \rho)$ on the vertices of C. If C has two vertices, then $\mathcal{S}(C, \rho)$ contains a single edge incident to them. Otherwise, the edges of $\mathcal{S}(C, \rho)$ are the edges of the 2-breaks in ρ.*

See Figure 3.9 a) and b) for an example.

Definition 44 (Scenario matching)**.** *A parsimonious 2-break scenario ρ for a circle C transforms it into a terminal graph that has equal sets of black and gray edges. Delete its gray edges to obtain a perfect matching of the scenario graph $\mathcal{S}(C, \rho)$, that we denote by $\mathcal{M}(C, \rho)$.*

Observation 19. *The scenario matching of a parsimonious black-2-break scenario for a circle consists of the edges that are gray in the circle.*

Definition 45 (Circular straight-line drawing of a scenario graph)**.** *A circular straight-line drawing of a graph is a drawing on a plane with the vertices of the graph arranged on a circle and the edges drawn as straight lines. If the edges in the drawing do not cross, then the drawing is an* embedding*. Fix a circular straight-line embedding Σ_C of a circle C. The scenario graph $\mathcal{S}(C, \rho)$ for a parsimonious 2-break scenario ρ for C also inherits a circular straight-line drawing $\Sigma_{\mathcal{S}(C,\rho)}$ from Σ_C.*

See Figure 3.9 for an example of a scenario graph, scenario matching, and their circular straight-line embeddings. In what follows we will suppose that a circle C comes together with a circular straight-line embedding Σ_C.

Definition 46 (Planar graph)**.** *A graph is* planar *if it admits an embedding. An embedding divides the plane into regions, called* faces*. One of them is* unbounded *or* infinite*, while the rest are* bounded*.*

Theorem 7. *Take a parsimonious 2-break scenario ρ for a circle C. The scenario graph $\mathcal{S}(C, \rho)$ is planar and all the bounded faces of its embedding $\Sigma_{\mathcal{S}(C,\rho)}$ are quadrilaterals. In addition to that, there exists a bijection between the 2-breaks in ρ and the bounded faces of $\Sigma_{\mathcal{S}(C,\rho)}$, that associates to a 2-break in ρ the face bounded by its edges.*

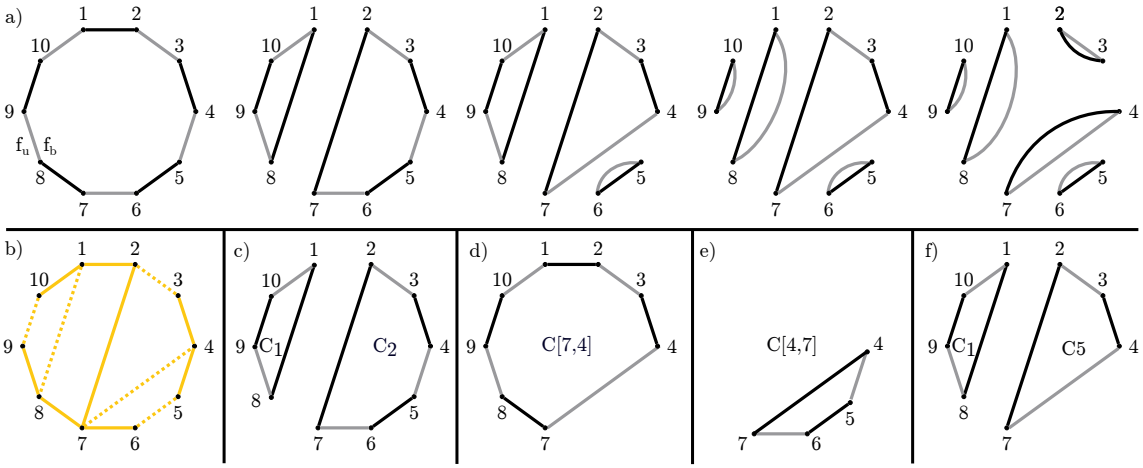


Figure 3.9: A circular straight-line embedding Σ_C of a circle C is depicted in a) on the left. This embedding divides the plane into two faces. f_u is unbounded, while f_b is bounded. The rest of a) depicts a parsimonious 2-break scenario $\rho = \Big(\Big(\big\{\{1,2\},\{7,8\}\big\} \rightarrow \big\{\{1,8\},\{2,7\}\big\}, black\Big), \Big(\big\{\{4,5\},\{6,7\}\big\} \rightarrow \big\{\{4,7\},\{5,6\}\big\}, gray\Big), \Big(\big\{\{1,10\},\{8,9\}\big\} \rightarrow \big\{\{1,8\},\{9,10\}\big\}, gray\Big), \Big(\big\{\{3,4\},\{2,7\}\big\} \rightarrow \big\{\{2,3\},\{4,7\}\big\}, black\Big)\Big)$ for C. The circular straight-line embedding $\Sigma_{\mathcal{S}(C,\rho)}$ of a scenario graph $\mathcal{S}(C,\rho)$ inherited from Σ_C is depicted in b). The edges of the scenario graph that are in the scenario matching $\mathcal{M}(C,\rho)$ appear dashed in b). The first 2-break τ in ρ transforms C into a union of two circles C_1 and C_2 of C depicted in c). According to the notation to be presented in Subsection 3.5.3, C_1 and C_2 are the sub-circles $C[8,1]$ and $C[2,7]$ of C. What follows is an example for Lemma 19 and Lemma 30 with $i = 4$ and $j = 7$. The sub-circles $C[7,4]$ and $C[4,7]$ are depicted in d) and e). The first 2-break τ in ρ transforms $C[7,4]$ into a union of two vertex-disjoint circles C_1 and C_5 as depicted in f). ρ can be partitioned into 2-break scenarios $\rho_{[4,7]}$ and $\rho_{[7,4]}$ for the sub-circles $C[4,7]$ and $C[7,4]$, with $\rho_{[4,7]} = \Big(\Big(\big\{\{4,5\},\{6,7\}\big\} \rightarrow \big\{\{4,7\},\{5,6\}\big\}, gray\Big)\Big)$, and $\rho_{[7,4]} = \Big(\Big(\big\{\{1,2\},\{7,8\}\big\} \rightarrow \big\{\{1,8\},\{2,7\}\big\}, black\Big), \Big(\big\{\{1,10\},\{8,9\}\big\} \rightarrow \big\{\{1,8\},\{9,10\}\big\}, gray\Big), \Big(\big\{\{3,4\},\{2,7\}\big\} \rightarrow \big\{\{2,3\},\{4,7\}\big\}, black\Big)\Big)$.

Proof. The proof is by induction on the number of vertices in C. C has an even number of vertices by construction. If C has two vertices, then ρ is empty, its scenario graph has a single edge and no bounded faces. Suppose that the statement is true for every circle having at most $2k-2 \geq 2$ vertices and take a circle C with $2k$ vertices and its parsimonious 2-break scenario ρ.

Due to Corollary 2, the first 2-break $\Big(\big\{\{u,v\},\{w,s\}\big\} \to \big\{\{u,w\},\{v,s\}\big\}, col\Big)$ of ρ transforms C into a union of two vertex disjoint circles. Denote them by C_1 and C_2. Due to Corollary 4, the rest of ρ can be partitioned into ρ_1 and ρ_2, that are 2-break scenarios for C_1 and C_2 respectively. The edges of $\Sigma_{\mathcal{S}(C,\rho)}$ can be obtained by adding the edges $\{u,v\}$ and $\{w,s\}$ to the union of the edges of $\mathcal{S}(C_1,\rho_1)$ and $\mathcal{S}(C_2,\rho_2)$. $\Sigma_{\mathcal{S}(C_1,\rho_1)}$ and $\Sigma_{\mathcal{S}(C_2,\rho_2)}$ satisfy the inductive hypothesis, thus their edges do not cross. $\{u,v\}$ and $\{w,s\}$ do not cross the other edges and together with $\{u,w\}$ and $\{v,s\}$ bound the only face of $\Sigma_{\mathcal{S}(C,\rho)}$ not belonging to $\Sigma_{\mathcal{S}(C_1,\rho_1)}$ or $\Sigma_{\mathcal{S}(C_2,\rho_2)}$. □

See Figure 3.10 for an illustration of the links between the scenario and the trajectory graphs.

3.5.3 Partitioning a Parsimonious 2-break Scenario for a Circle into the Scenarios for its Sub-circles

In this section we introduce a notion of a *sub-circle of a circle* and demonstrate that a parsimonious 2-break scenario for a circle can be partitioned into parsimonious 2-break scenarios for its sub-circles.

Definition 47 (Sub-circle)**.** *Take an odd length path in a circle. If the edges incident to its endpoints are black (respectively gray), then join its endpoints with a gray (respectively black) edge to obtain a circle. The added colored edge is the* colored outer edge of the sub-circle.

Fix a circular straight-line embedding Σ_C of a circle C on n vertices. Number them $[\![1,n]\!]$ while respecting their clockwise order on Σ_C and ensuring that the colored edge going clockwise from n to 1 is gray. For $i,j \in [\![1,n]\!]$ with i and j of different parity, define $C[i,j]$ to be the sub-circle of C consisting of the path going clockwise from i to j in Σ_C and the colored outer edge. By construction, it is $(\{i,j\}, gray)$ if i is odd and $(\{i,j\}, black)$ otherwise. Denote this color by $col_{\{i,j\}}$, and the opposite color by $\overline{col_{\{i,j\}}}$.

Lemma 19. *Take a parsimonious 2-break scenario ρ for a circle C and an edge $\{i,j\}$ of its scenario graph. ρ can be partitioned into parsimonious 2-break scenarios for $C[i,j]$ and $C[j,i]$.*

Proof. The proof is by induction of the number of vertices in C. See Figure 3.9 for an example. If C has two vertices, then the scenario graph of ρ contains a single

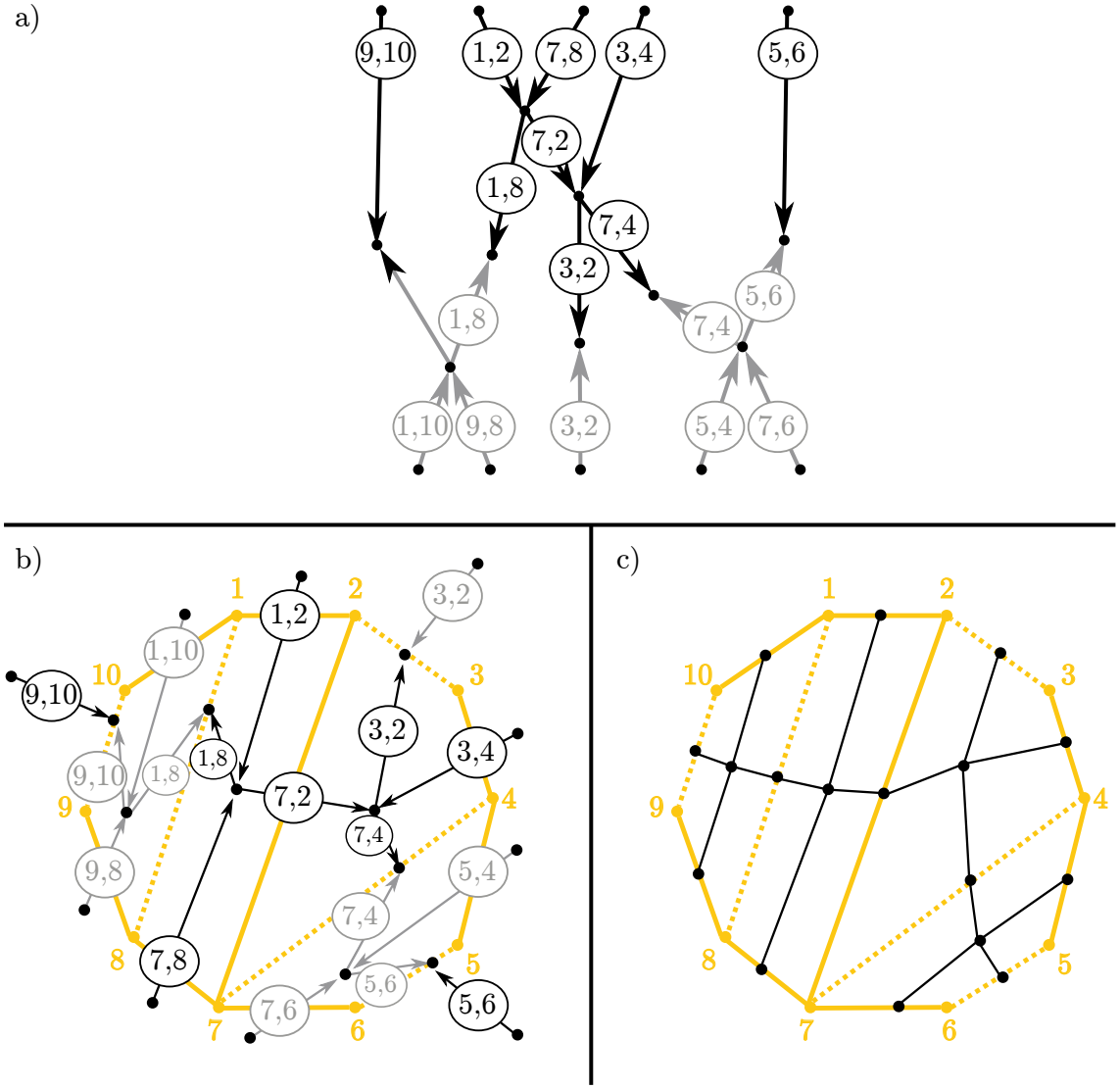


Figure 3.10: An embedding of a trajectory graph $\mathcal{D}(C, \rho)$ for a circle C and its parsimonious 2-break scenario ρ from Figure 3.9 is depicted in a). b) presents another embedding of $\mathcal{D}(C, \rho)$, where it is drawn over the embedding $\Sigma_{\mathcal{S}(C,\rho)}$ of the scenario graph from Figure 3.9. To each edge and face of $\Sigma_{\mathcal{S}(C,\rho)}$ assign a vertex as depicted in c), and join two vertices if they correspond to a face incident to an edge. The ternary tree thus obtained is, up to minor modifications, equal to $\mathcal{D}(C, \rho)$ once the edge labels, colors and orientations are forgotten.

edge $\{1,2\}$, while ρ and the parsimonious 2-break scenarios for $C[1,2]$ and $C[2,1]$ are empty. Suppose that the statement is true for any circle with at most $2k-2 \geq 2$ vertices and take a circle C on $2k$ vertices. The first 2-break τ in ρ transforms C into a union of two vertex disjoint circles due to Corollary 2. Denote them by C_1 and C_2. Due to Corollary 4, the rest of ρ can be partitioned into parsimonious 2-break scenarios ρ_1 and ρ_2 for C_1 and C_2.

Due to Theorem 7, the scenario graph of ρ inherits from Σ_C a circular straight-line embedding in which all the bounded faces are quadrilaterals. The edges of τ and the edge $\{i,j\}$ all belong to the scenario graph of ρ. This means that i and j belong either to C_1 or C_2, without loss of generality we can suppose that it is C_2. This also means that all the vertices of τ belong either to $C[i,j]$ or $C[j,i]$. Without loss of generality we can suppose that it is $C[j,i]$.

The inductive hypothesis holds for the triplet $(C_2, \rho_2, \{i,j\})$, providing us with a partition of ρ_2 into two parsimonious 2-break scenarios for the sub-circles of (C_2, λ_2) having $(\{i,j\}, black)$ and $(\{i,j\}, gray)$ as the colored outer edges. By construction, one of these labeled sub-circles is $C[i,j]$. Denote the other one by C_5, and denote the parsimonious 2-break scenarios obtained for them by $\rho_{[i,j]}$ and ρ_5 respectively.

By now we have established that ρ is a shuffle of τ, ρ_1, $\rho_{[i,j]}$, and ρ_5. Remove $\rho_{[i,j]}$ and τ from ρ to obtain a shuffle of ρ_1 and ρ_5, that we denote by $\overline{\rho}$. Due to Observation 15, $\overline{\rho}$ is a parsimonious 2-break scenario for the union of the vertex-disjoint circles C_1 and C_5, which is also the graph obtained from $C[j,i]$ after τ is performed. Finally, by deleting only $\rho_{[i,j]}$ from ρ, we obtain a shuffle of τ and $\overline{\rho}$, that is a parsimonious 2-break scenario for $C[j,i]$, that together with $\rho_{[i,j]}$ satisfies the statement. □

Definition 48 (Matched scenario for a sub-circle)**.** *A parsimonious 2-break scenario for a sub-circle $C[i,j]$ is* matched*, if its scenario matching includes the edge $\{i,j\}$. It is* non-matched *otherwise.*

Theorem 8. *Take a matched parsimonious 2-break scenario ρ for $C[i,j]$ with $i+3 \leq j$. ρ can be partitioned into a 2-break $\tau = \left(\left\{\{i,k\},\{l,j\}\right\} \to \left\{\{i,j\},\{k,l\}\right\}, \overline{col_{\{i,j\}}}\right)$ with $i < k < l < j$, a parsimonious 2-break scenario for $C[k,l]$ and matched parsimonious 2-break scenarios for $C[i,k]$ and $C[l,j]$.*

Proof. The scenario graph $\mathcal{S}(C,\rho)$ inherits its embedding $\Sigma_{\mathcal{S}(C,\rho)}$ from Σ_C. An edge $\{i,j\}$ is incident to a single bounded face of the embedding $\Sigma_{\mathcal{S}(C,\rho)}$, thus due to Theorem 7 there exists a single 2-break τ in ρ with $\{i,j\}$ among its edges. τ introduces a colored edge $(\{i,j\}, \overline{col_{\{i,j\}}})$, since ρ is matched. This means that there exist $i < k < l < j$ such that $\tau = \left(\left\{\{i,k\},\{l,j\}\right\} \to \left\{\{i,j\},\{k,l\}\right\}, \overline{col_{\{i,j\}}}\right)$. The scenario graph of ρ contains the edges $\left\{\{i,j\},\{i,k\},\{k,l\},\{l,j\}\right\}$ of τ. Applying Lemma 19 for $\{i,k\},\{k,l\}$ and $\{l,j\}$, we obtain parsimonious 2-break scenarios

ρ_1, ρ_2, and ρ_3 for the sub-circles $C[i,k]$, $C[k,l]$ and $C[l,j]$ that together with τ partition ρ. All we have to prove now is that ρ_1 and ρ_3 are matched.

If $k = i+1$, then $C[i,k]$ has two vertices and ρ_1 is matched by definition. Thus we can suppose that $i+3 \leq k$, which means that $(\{i,k\}, \overline{col_{\{i,j\}}})$ is not present in $C[i,j]$. τ replaces this colored edge, thus due to Observation 1 ρ also contains a 2-break introducing this edge. Due to Theorem 7 there are only two 2-breaks in ρ with $\{i,k\}$ among their edges. Due to the same theorem ρ_1 contains a 2-break with $\{i,k\}$ among its edges. By construction τ is not in ρ_1, thus we can conclude that ρ_1 contains a 2-break introducing $(\{i,k\}, \overline{col_{\{i,j\}}})$, which means that ρ_1 is matched. The same analysis applies to ρ_3. □

Analogous arguments to those presented in proof of Theorem 8 allow us to prove the following.

Theorem 9. *Take a non-matched parsimonious 2-break scenario ρ for $C[i,j]$ with $i+3 \leq j$. ρ contains a 2-break $\tau = \left(\left\{\{i,j\},\{k,l\} \to \left\{\{i,k\},\{l,j\}\right\}\right\}, col_{\{i,j\}}\right)$ with $i < k < l < j$, and ρ can be partitioned into τ, parsimonious 2-break scenarios for $C[i,k]$ and $C[l,j]$, and a parsimonious matched 2-break scenario for $C[k,l]$.*

These two theorems allow us to partition a parsimonious 2-break scenario for a sub-circle of C into parsimonious 2-break scenarios for smaller sub-circles of C.

3.6 A Bijection Between the Equivalence Classes of the Parsimonious 2-break Scenarios for a Circle and its Matched Quadrangulations

3.6.1 Introduction

In Theorem 7 we have established that the scenario graph of a parsimonious 2-break scenario for a circle is a quadrangulation of a regular polygon. We define 2-break scenarios to be *equivalent* if their multisets of 2-breaks are equal and establish a bijection between the equivalence classes of the parsimonious 2-break scenarios for a circle and the *matched quadrangulations* of a regular polygon.

3.6.2 The Scenario Graphs and Matchings of Equivalent Scenarios are Equal

Definition 49 (Equivalent 2-break scenarios)**.** *Two 2-break scenarios are* equivalent *if their multisets of 2-breaks are equal.*

Definition 50 (Edge count of a 2-break scenario)**.** *Take a pair of vertices $\{u, v\}$, a color col and a 2-break τ. If τ replaces one colored edge (respectively two colored edges) $(\{u, v\}, col)$, then its $(\{u, v\}, col)$-count is -1 (respectively -2). If τ introduces one colored edge (respectively two colored edges) $(\{u, v\}, col)$, then its $(\{u, v\}, col)$-count is 1 (respectively 2). $(\{u, v\}, col)$-count of a 2-break scenario is the sum of the $(\{u, v\}, col)$-counts of its 2-breaks.*

Lemma 20. *Equivalent 2-break scenarios for a graph transform it into equal terminal graphs.*

Proof. Take a graph G, a 2-break scenario ρ transforming G into a terminal graph $\overline{G}$, and two vertices $\{u, v\}$ together with a color *col*. To the number of colored edges $(\{u, v\}, col)$ in G add the $(\{u, v\}, col)$-count of ρ to obtain the number of colored edges $(\{u, v\}, col)$ in $\overline{G}$. This quantity stays the same for an equivalent 2-break scenario, meaning that it also transforms G into a terminal graph $\overline{G}$. □

Lemma 21. *Take a parsimonious 2-break scenario ρ for a circle C and a 2-break τ in ρ replacing two colored edges of C. The sequence of 2-breaks with τ performed first and followed by the rest of ρ is a parsimonious 2-break scenario for C.*

Proof. Without loss of generality we can suppose that τ does not replace the colored edge $(\{1, n\}, gray)$. In this case there exists $i + 3 \leq j$ and a color *col*, such that $\tau = \Big(\Big\{\{i, i+1\}, \{j, j+1\}\Big\} \to \Big\{\{i, j+1\}, \{i+1, j\}\Big\}, col\Big)$. Due to Theorem 7, there exists a face in the embedding of the scenario graph $\Sigma_{\mathcal{S}(C,\rho)}$ bounded by the edges of τ. Applying Lemma 19 twice we obtain that ρ can be partitioned into parsimonious 2-break scenarios for $C[j + 1, i]$, $C[i + 1, j]$ and τ. Due to Corollary 4, ρ without τ is a is a parsimonious 2-break scenario for the union of $C[j + 1, i]$ and $C[i + 1, j]$, that is a graph obtained from C after τ is performed. This means that moving τ to the beginning of ρ we obtain a parsimonious 2-break scenario for C. □

Theorem 10. *The scenario graphs and matchings of two parsimonious 2-break scenarios for a circle are equal if and only if these scenarios are equivalent.*

Proof. The proof is by induction on the number of vertices in a circle C. If it has two vertices, then its parsimonious 2-break scenario is empty and the statement is true. Suppose that the statement is true for every circle with at most $2k - 2 \geq 2$ vertices and take a circle C on $2k$ vertices together with its parsimonious 2-break scenarios ρ^1 and ρ^2.

First suppose that ρ^1 and ρ^2 are equivalent. This means that they contain exactly the same 2-breaks but possibly in different order. By definition, the edges of $\Sigma_{\mathcal{S}(C,\rho^1)}$ are the edges of the 2-breaks in ρ^1, and these are exactly the same as the edges of the 2-breaks in ρ^2, meaning that $\Sigma_{\mathcal{S}(C,\rho^1)} = \Sigma_{\mathcal{S}(C,\rho^2)}$. Due to Lemma 20, the terminal graphs provided by ρ^1 and ρ^2 are equal. This means that their scenario matchings are also equal.

Now suppose that the scenario graphs and the scenario matchings of ρ^1 and ρ^2 are equal. Take the first 2-break τ^1 of ρ^1. Without loss of generality we can suppose that τ^1 does not replace $(\{1, n\}, gray)$. In this case there exists $i < j$ and a color col, such that $\tau^1 = \Big(\Big\{\{i, i+1\}, \{j, j+1\}\Big\} \to \Big\{\{i, j+1\}, \{i+1, j\}\Big\}, col\Big)$. As $(\{i, i+1\}, col)$ is an edge of C, there is a single bounded face of $\Sigma_{\mathcal{S}(C,\rho^1)}$ incident to $\{i, i+1\}$. Due to Theorem 7, τ^1 is the single 2-break in ρ^1 with $\{i, i+1\}$ among its edges. This means that $\{i, i+1\}$ is not in the scenario matching of ρ^1 and thus neither in that of ρ^2.

Due to Theorem 7, there exists a face in $\Sigma_{\mathcal{S}(C,\rho^1)}$ bounded by the edges of τ^1. $\Sigma_{\mathcal{S}(C,\rho^1)} = \Sigma_{\mathcal{S}(C,\rho^2)}$, thus the latter contains the same face that we denote by f. Due to Theorem 7, ρ^2 contains a 2-break τ^2 with the edges of τ^2 bounding f, and τ^2 being the single 2-break in ρ^2 with $\{i, i+1\}$ among its edges. $\{i, i+1\}$ is not present in the scenario matching of ρ^2, and the colored edge $(\{i, i+1\}, col)$ is present in C, thus τ^2 must replace this colored edge, meaning that $\tau^2 = \tau^1$, denote them by τ.

Due to Lemma 21, τ can be moved to the beginning of ρ^2 to obtain an equivalent parsimonious 2-break scenario ρ^3. Due to Corollary 2, τ transforms C into a union of vertex disjoint circles, denote them by C_1 and C_2. Due to Corollary 4, the rest of ρ^1 (respectively ρ^3) can be partitioned into two sub-sequences ρ^1_1 and ρ^1_2 (respectively ρ^3_1 and ρ^3_2) that are parsimonious 2-break scenarios for C_1 and C_2. The scenario graphs and matchings of ρ^1_1 and ρ^3_1 are equal, and so are those of ρ^1_2 and ρ^3_2. Thus ρ^1_1 and ρ^3_1 are equivalent, due to the induction hypothesis, and so are ρ^1_2 and ρ^3_2. This allows us to conclude that ρ^1 and ρ^3 are equivalent, and thus are ρ^1 and ρ^2. □

Corollary 6. *Due to Observation 19, the scenario matchings of two parsimonious black-2-break scenarios are equal by construction, thus they are equivalent if and only if their scenario graphs are equal.*

3.6.3 A Matched Quadrangulation

Definition 51 (Matched quadrangulation of Σ_C)**.** *Take a circular straight-line embedding Σ_C of a circle C and forget the colors of its colored edges to obtain a regular polygon. If C has two vertices, then keep a single edge joining them to obtain an embedding Σ_S. Otherwise, add straight-line edges to obtain an embedding Σ_S with all the inner faces being quadrilaterals. We call this embedding a* quadrangulation of a regular polygon*. A pair of Σ_S and its perfect matching $\mathcal{M}$ is a* matched quadrangulation of Σ_C*.*

See Figure 3.11 for an example of a matched quadrangulation and an example for Theorem 11.

Theorem 11. *Take a circle C and its matched quadrangulation $(\Sigma_S, \mathcal{M})$. There exists a parsimonious 2-break scenario ρ for which $\Sigma_{\mathcal{S}(C,\rho)} = \Sigma_S$ and $\mathcal{M}(C, \rho) = \mathcal{M}$.*

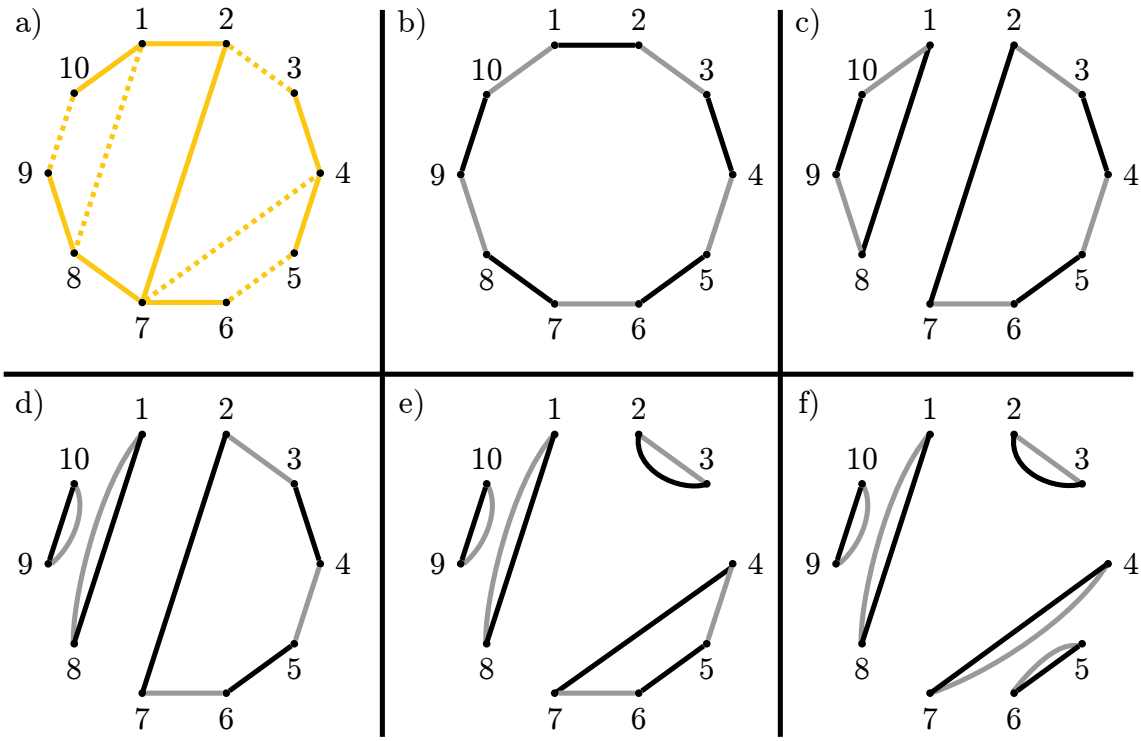


Figure 3.11: A matched quadrangulation $(\Sigma_S, \mathcal{M})$ of a circular straight-line embedding of a circle is depicted in a) with the matched edges indicated in dashed lines. Following the process described in Theorem 11, we obtain a parsimonious 2-break scenario ρ with the scenario graph and matching of ρ being equal to Σ_S and $\mathcal{M}$. ρ is depicted in b)-f). To begin with, Σ_S contains a face bounded by the edges $\{1,2\},\{2,7\},\{7,8\}$ and $\{8,1\}$, with $(\{1,2\}, black)$ and $(\{7,8\}, black)$ being the colored edges of C. $\{1,2\}$ and $\{7,8\}$ do not belong to $\mathcal{M}$, thus we start ρ with a 2-break $\Big(\big\{\{1,2\},\{2,7\}\big\} \rightarrow \big\{\{1,7\},\{2,8\}\big\}, black\Big)$ that leaves us with two subcircles $C[8,1]$ and $C[2,7]$, for which we proceed with the same process. Σ_S contains a face bounded by the edges $\{1,8\},\{8,9\},\{9,10\}$ and $\{10,1\}$, with $(\{9,10\}, black)$ and $(\{1,8\}, black)$ being the colored edges of $C[8,1]$. $\{9,10\}$ and $\{1,8\}$ belong to $\mathcal{M}$, thus we proceed with a 2-break $\Big(\big\{\{10,1\},\{8,9\}\big\} \rightarrow \big\{\{9,10\},\{1,8\}\big\}, gray\Big)$.

Proof. The proof is by induction on the number of vertices in C. If C has 2 vertices, then the empty scenario satisfies the statement. Suppose that the statement is true for every circle having at most $2k-2 \geq 2$ vertices and take a circle on $2k$ vertices.

Due to Theorem 7 and Theorem 1, Σ_S has $k-1$ bounded face. On the other hand, Σ_S has k edges of the form $\{i, i+1\}$ that are black in C, and such an edge is incident to at most one bounded face of Σ_S. This way we obtain that Σ_S contains a pair of edges that are black in C incident to the same bounded face f. Denote these black edges by $(\{i, i+1\}, black)$ and $(\{j, j+1\}, black)$ with $i < j$.

All the edges in Σ_S, except $\{i, i+1\}$ and $\{j, j+1\}$, are incident to an even number of vertices of $C[i+1, j]$, which itself has an even number of vertices. Thus either both $\{i, i+1\}$ and $\{j, j+1\}$ are in $\mathcal{M}$, or both are not in $\mathcal{M}$.

Denote the subgraphs of Σ_S induced by the vertices of $C[i+1, j]$ and $C[j+1, i]$ by Σ_{S_1} and Σ_{S_2} respectively. Denote by $\mathcal{M}_1$ (respectively $\mathcal{M}_2$) a subset of $\mathcal{M}$ consisting of the edges with both ends in $C[i+1, j]$ (respectively $C[j+1, i]$).

Suppose that $\{i, i+1\}$ and $\{j, j+1\}$ are not in $\mathcal{M}$. In this case $\mathcal{M}_1$ and $\mathcal{M}_2$ partition $\mathcal{M}$ and are the matchings of Σ_{S_1} and Σ_{S_2}. Due to the induction hypothesis, there exist parsimonious 2-break scenarios ρ_1 and ρ_2 for $C[i+1, j]$ and $C[j+1, i]$ satisfying the statement of the theorem. A 2-break $\Big(\Big\{\{i, i+1\}, \{j, j+1\}\Big\} \rightarrow \Big\{\{i, j+1\}, \{i+1, j\}\Big\}, black\Big)$ followed by ρ_1 and ρ_2 is a parsimonious 2-break scenario for C satisfying the statement of the theorem.

Suppose that $\{i, i+1\}$ and $\{j, j+1\}$ are in $\mathcal{M}$. In this case the vertices $i+1$ and j are not incident to any edges in $\mathcal{M}_1$ and the vertices $j+1$ and i are not incident to any edges in $\mathcal{M}_2$. Add $\{i+1, j\}$ to $\mathcal{M}_1$ and $\{j+1, i\}$ to $\mathcal{M}_2$ to obtain the matchings $\mathcal{M}_1'$ of Σ_{S_1} and $\mathcal{M}_2'$ of Σ_{S_2}. Due to the induction hypothesis, there exist parsimonious 2-break scenarios ρ_1 and ρ_2 for $C[i+1, j]$ and $C[j+1, i]$, satisfying the statement of the theorem. ρ_1 and ρ_2 followed by a 2-break $\Big(\Big\{\{i, j+1\}, \{i+1, j\}\Big\} \rightarrow \Big\{\{i, i+1\}, \{j, j+1\}\Big\}, gray\Big)$ is a parsimonious 2-break scenario for C satisfying the statement of the theorem. □

Corollary 7. *Take a quadrangulation of a regular polygon Σ_S and its matching $\mathcal{M}_g$ containing the edges that are gray in C. There exists a parsimonious black-2-break scenario ρ for which $\Sigma_{\mathcal{S}(C,\rho)} = \Sigma_S$ and $\mathcal{M}(C, \rho) = \mathcal{M}_g$.*

3.6.4 A Bijection Between the Equivalence Classes of the Parsimonious 2-break Scenarios and the Equivalence Classes of the MLTDs

In Section 2.4 we have established a bijection between the parsimonious black-2-break scenarios for the permutation breakpoint graph $H(\sigma, id)$ and the MLTDs of σ. Here we use some classical results concerning the equivalence classes of the

MLTDs of a cyclic permutation σ to establish a bijection between them and the equivalence classes of the parsimonious black-2-break scenarios for $H(\sigma, id)$.

Definition 52 (Equivalent transposition decompositions)**.** *Two transposition decompositions of a permutation are equivalent if their multisets of transpositions are equal.*

Lemma 22. *Take two equivalent parsimonious black-2-break scenarios for $H(\sigma, id)$. The MLTDs obtained for them in Lemma 8 are also equivalent.*

Proof. Take a 2-break τ in a parsimonious black-2-break scenario ρ for $H(\sigma, id)$. Due to Lemma 8, τ is a preserving 2-break and thus of the form $\Big(\big\{\{u_h, v_t\}, \{w_h, s_t\}\big\} \rightarrow \big\{\{u_h, s_t\}, \{w_h, v_t\}\big\}, black\Big)$ for some $u, v, w, s \in \{1, \ldots, n\}$. The bijection in Lemma 8 works by transforming τ into its transposition $\pi(\tau) = (vs)$. The multisets of 2-breaks of two equivalent black-2-break scenarios are equal, this means that the multisets of transpositions of the MLTDs obtained for them in Lemma 8 are also equal. □

Using Lemma 22 we obtain a function F that assigns an equivalence class of the MLTDs to an equivalence class of the parsimonious black-2-break scenarios. The way that we assign a parsimonious black-2-break scenario for $H(\sigma, id)$ to a given MLTD of σ in Lemma 8 relies on the order of transpositions in the MLTD, thus it is not immediately clear if F is a bijection. To prove this, we have to use the machinery introduced by Dulucq and Penaud [38] and later used by Farnoud and Milenkovic [41].

Definition 53 (The graph of an MLTD)**.** *For a permutation $\sigma \in \mathbb{S}_n$ and its MLTD T define a 1-edge-colored graph $\mathcal{T}(T) = (V, E)$, with $V = \{1, \ldots, n\}$ and $E = \{\{i, j\} | (i, j) \in T\})$.*

Definition 54 (Planar graph on a circle)**.** *Embed vertices $\{1, \ldots, n\}$ on a circle while respecting their order. If a straight line drawing of a graph $\mathcal{T}$ on these vertices is an embedding, then $\mathcal{T}$ is a* planar graph on a circle*.*

See Figure 3.12 d) and e) for an example of a planar spanning tree on a circle.

Theorem 12 (Dulucq and Penaud [38], Farnoud and Milenkovic [41])**.** *Take an MLTD T of a cyclic permutation $\sigma \in \mathbb{S}_n$. $\mathcal{T}(T)$ is a planar spanning tree on a circle. Now, take a planar spanning tree $\mathcal{T}$ on a circle. There exists an MLTD T' of σ satisfying $\mathcal{T}(T') = \mathcal{T}$. There are $\frac{1}{2n+1}\binom{3n}{n}$ planar spanning trees on a circle with $n + 1$ vertices, thus this is also the number of the equivalence classes of the MLTDs of a cyclic permutation of $n + 1$ elements.*

A sequence with values $\frac{1}{2n+1}\binom{3n}{n}$ is called the Fuss-Catalan sequence. As Catalan numbers count the number of triangulations of a regular polygon, so the Fuss-Catalan numbers count the number of quadrangulations of a regular polygon.

Lemma 23 (Baryshnikov [8])**.** *The number of quadrangulations of a regular polygon on* $2n+2$ *vertices is equal to* $\frac{1}{2n+1}\binom{3n}{n}$.

Corollary 6 and Corollary 7 allow us to conclude with a following lemma.

Lemma 24. *The number of equivalence classes of the black-2-break scenarios for a circle on* $2n$ *vertices is equal to the number of quadrangulations of a regular polygon on* $2n$ *vertices.*

Due to Theorem 12, Lemma 23 and Lemma 24, there is an equal number of the equivalence classes of MLTDs of a cyclic permeation σ and the equivalence classes of the parsimonious black-2-break scenarios for the permutation breakpoint graph $H(\sigma, id)$. Apostolakis in Section 3.3 of [5] presents a bijection between the quadrangulations, that he calls *quadrangular dissections*, and the planar spanning trees on a circle, that he calls *non-crossing trees*. The construction provided in [5] and presented in Figure 3.12 establishes that F is indeed a bijection between these equivalence classes.

3.7 Conclusion

We have shown that a parsimonious 2-break scenario for a graph can be partitioned into parsimonious 2-break scenarios for the circles of its simple cycles. Theorem 8 allows us to partition a parsimonious 2-break scenario for a circle into parsimonious 2-break scenarios for its sub-circles. In Chapter 5 this will lead to a dynamic programming algorithm for exploring the space of the parsimonious 2-break scenarios for a circle. Taken together these results allow us to explore the space of the parsimonious 2-break scenarios for any graph.

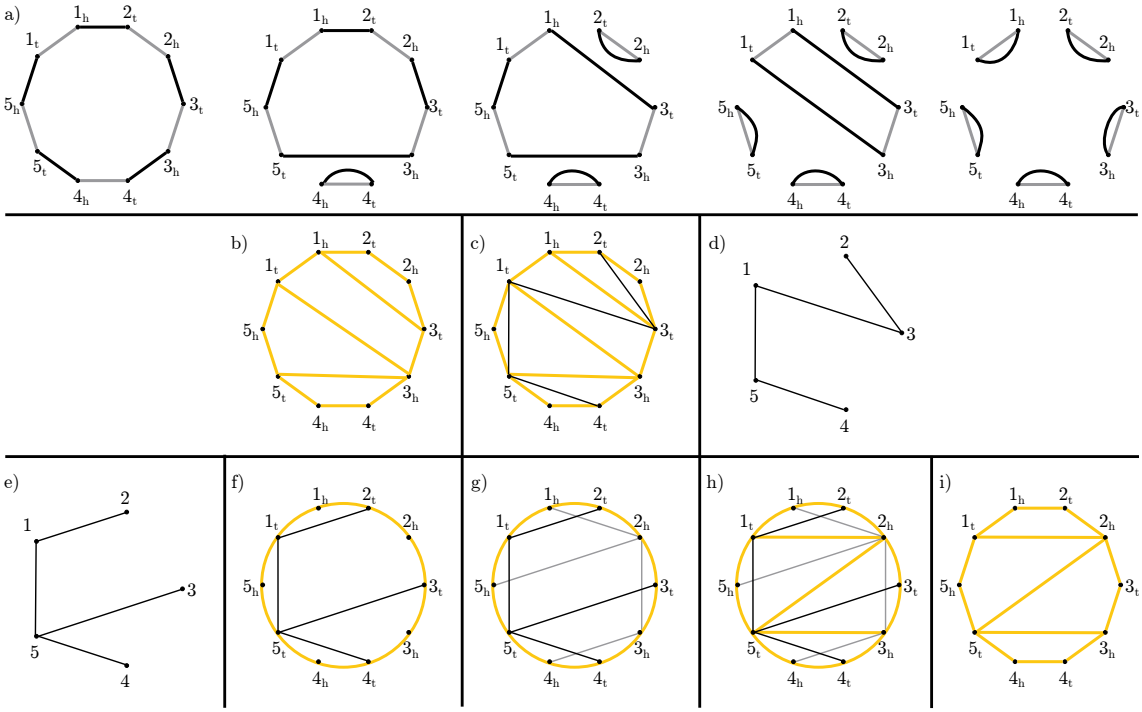


Figure 3.12: A black-2-break scenario ρ for the permutation breakpoint graph $H((12345), id)$ is depicted in a), while its scenario graph is presented in b). Using Lemma 8, ρ provides us with a Minimum Length Transposition Decomposition $T = ((45), (23), (15), (13))$ of (12345). $((23), (45), (15), (13))$ is an equivalent MLTD to T. Graphically the transpositions of T can be interpreted as the diagonals of the bounded faces of the scenario graph joining the *tail* vertices, as depicted in c). These diagonals correspond to a planar spanning tree on a circle $\mathcal{T}(T)$ given in d). $\mathcal{T}$, another planar spanning tree on a circle, is depicted in e). We provide in i) a quadrangulation of a regular polygon corresponding to an equivalence class $\mathcal{E}$ of the parsimonious black-2-break scenarios for $H((12345), id)$, for which $F(\mathcal{E})$ is the equivalence class of the MLTDs of σ corresponding to $\mathcal{T}$. $\mathcal{T}$ partitions a circle into 5 regions and each region corresponds to an arc of the circle. To each arc/region assign a *head* vertex as depicted in f). Add a gray edge between the head vertices if their corresponding regions are separated by an edge of $\mathcal{T}$. The obtained graph $\kappa(\mathcal{T})$ is called the *complement* of $\mathcal{T}$ in [5]. Each edge of $\mathcal{T}$ intersects exactly one edge of $\kappa(\mathcal{T})$ and vice versa, the pairs of the intersecting edges correspond to the diagonals of the quadrilaterals depicted in h) and i).

Chapter 4

Cost Constrained 2-break Scenarios

4.1 Introduction

In this chapter we present our model for assigning costs to 2-break scenarios and introduce the problem of finding a minimum cost scenario among those of minimum length, the φ-Minimum Cost Parsimonious Scenario problem, that will be treated in Chapter 5.

In [95] we proposed a method to cost constrain DCJs based on the spatial proximity of the intergenic regions, as it is one of the biological triggers for genome rearrangements [71]. We labeled edges of the genome breakpoint graph with the spatial positions of the corresponding genome adjacencies, and allowed a 2-break to transform two edges labeled x and y into 2 new edges with the same labels. Within this model the cost of a 2-break is 0 if the labels are equal and 1 otherwise. We posed the Minimum Local Parsimonious Scenario problem (MLPS) of finding a parsimonious 2-break scenario minimizing the sum of the costs of its 2-breaks.

Around the same time Bulteau, Fertin, and Tannier [27] proposed a method to cost constrain DCJs based on the lengths of the intergenic regions. Here once again edges of the breakpoint graph were labeled, however an additional operation, that of changing an edge label, was allowed. The Sorting by wDCJs and Indels in Intergenes problem (presented in Subsection 4.3.2), asked for an optimal 2-break scenario among the parsimonious ones. Bulteau, Fertin, and Tannier [27] proposed a polynomial time algorithm solving this problem for a circle.

In [95] we also started our work on MLPS by providing a polynomial time algorithm solving it for a circle. We then proceeded by showing how this algorithm can be used as a sub-routine to solve MLPS for the genome breakpoint graph. We soon realized that our method could be extended beyond the breakpoint graphs,

and that we could incorporate the edge label changes proposed by Bulteau, Fertin, and Tannier [27].

In a seemingly different line of work Farnoud and Milenkovic [41] allowed for an arbitrary positive real valued cost function on transpositions and defined the cost of a transposition decomposition of a permutation to be the sum of the costs of its transpositions. The authors proposed an $O(n^4)$-time algorithm for finding a minimum cost decomposition among the Minimum Length Transposition Decompositions of a permutation $\sigma \in \mathbb{S}_n$.

These three lines of work converged into a framework for cost constrained 2-breaks outlined in Section 4.2 and first presented by us in [90]. The rest of the chapter consists of novel but rather technical work aimed to better understand the structure of the minimum cost scenarios.

4.2 An $\mathcal{O}$-scenario

In this section we augment a graph with labels on both vertices and edges, and introduce an $\mathcal{O}$-*scenario* transforming one labeled graph into another. The labeled edges can be modified either via an $\mathcal{O}$-*change*, that changes the label of a single edge, or via an $\mathcal{O}$-*break*, that is a 2-break acting not only on the connectivity of the edges but also on their labels. The labels of the vertices stay fixed throughout an $\mathcal{O}$-scenario, however they are used to define which $\mathcal{O}$-changes and $\mathcal{O}$-breaks are allowed to transform a labeled graph.

Take alphabets of vertex and edge labels Σ_V and Σ_E. We will use letters $\{u, v, w, s, r\}$ to denote vertices, $\{a, b, c, d\}$ to denote vertex labels and $\{x, y, z, t, q\}$ to denote edge labels. We start by introducing a set of valid operations $\mathcal{O}$.

Definition 55 (Valid operations $\mathcal{O}$)**.** *A* valid set of operations $\mathcal{O}$ *is a subset of tuples*

- $\Big((\{a, b\}, x); (\{a, b\}, y); col\Big)$ (*called* edge-label change)*, and*
- $\Big(\Big\{(\{a, b\}, x), (\{c, d\}, y)\Big\}; \Big\{(\{a, c\}, z), (\{b, d\}, t)\Big\}; col\Big)$ (*called* 2-break on labels)

for $a, b, c, d \in \Sigma_V$*,* $x, y, z, t \in \Sigma_E$*, and* $col \in \{$*gray,black*$\}$*.*

Having $\mathcal{O}$ defined on the labels of the vertices and not on the graph vertices themselves allows us to use it for any *labeled graph*.

Definition 56 (Labeled graph)**.** *A labeled graph is a pair of a graph* $G = (V, E)$ *and its labeling* $\lambda = (\lambda_V, \lambda_E)$ *with* $\lambda_V : V \to \Sigma_V$ *and* $\lambda_E : E \to \Sigma_E$*. Take a colored edge* $(\{u, v\}, col)$ *and its label* x *provided by* λ*, we denote this* labeled edge *by* $(\{u, v\}, col, x)$*.*

Definition 57 ($\mathcal{O}$-change)**.** *If a set of valid operations $\mathcal{O}$ contains an edge-label change $\Big((\{a,b\},x);(\{a,b\},y);col\Big)$ and (G,λ) contains a labeled edge $(\{u,v\},col,x)$, with vertices (u,v) labeled (a,b), then the label of this edge can be changed into y. We denote such a transformation of (G,λ) by $(\{u,v\},col,x)\to(\{u,v\},col,y)$, and call it an $\mathcal{O}$-change.*

Definition 58 ($\mathcal{O}$-break)**.** *If a set of valid operations $\mathcal{O}$ contains a 2-break on labels $\Big(\Big\{(\{a,b\},x),(\{c,d\},y)\Big\};\Big\{(\{a,c\},z),(\{b,d\},t)\Big\};col\Big)$ and (G,λ) contains labeled edges $(\{u,v\},col,x)$ and $(\{w,s\},col,y)$, with vertices (u,v,w,s) labeled (a,b,c,d) respectively, then a 2-break $\Big(\Big\{\{u,v\},\{w,s\}\Big\}\to\Big\{\{u,w\},\{v,s\}\Big\},col\Big)$ can be performed on G, with the labels of the new colored edges $(\{u,w\},col)$ and $(\{v,s\},col)$ being z and t. We call such a transformation of a labeled graph (G,λ) an $\mathcal{O}$-break and denote it $\Big(\Big\{(\{u,v\},x),(\{w,s\},y)\Big\}\to\Big\{(\{u,w\},z),(\{v,s\},t)\Big\},col\Big)$.*

Definition 59 ($\mathcal{O}$-scenario and its 2-break-length)**.** *An* $\mathcal{O}$-scenario *for (G,λ) is a sequence of $\mathcal{O}$-changes and $\mathcal{O}$-breaks transforming it into $(\overline{G},\overline{\lambda})$, with $\overline{G}$ being a terminal graph and its multisets of black and gray labeled edges being equal. The number of $\mathcal{O}$-breaks in an $\mathcal{O}$-scenario is its* 2-break-length*.*

See Figure 4.2 for an example of an $\mathcal{O}$-scenario. An $\mathcal{O}$-scenario does not necessarily exist for a given (G,λ), however if it exists, then the inequality $d_{\mathcal{O}b}(G,\lambda)\geq d_{2b}(G)$ holds, where $d_{\mathcal{O}b}(G,\lambda)$ denotes the minimum 2-break-length of an $\mathcal{O}$-scenario, and $d_{2b}(G)$ is the minimum length of a 2-break scenario for G.

Definition 60 (Parsimonious $\mathcal{O}$-scenario)**.** *An $\mathcal{O}$-scenario for (G,λ) is* parsimonious*, if its 2-break-length is exactly $d_{2b}(G)$. According to this definition a parsimonious $\mathcal{O}$-scenario can contain any number of $\mathcal{O}$-breaks.*

An $\mathcal{O}$-break $\Big(\Big\{(\{u,v\},x),(\{w,s\},y)\Big\}\to\Big\{(\{u,w\},z),(\{v,s\},t)\Big\},col\Big)$ can also be denoted by a pair of a 2-break $\tau=\Big(\Big\{\{u,v\},\{w,s\}\Big\}\to\Big\{\{u,w\},\{v,s\}\Big\},col\Big)$, and a labeling χ of its colored edges. We will allow ourselves to write $(\tau,\chi)=\Big(\Big\{(\{u,v\},x),(\{w,s\},y)\Big\}\to\Big\{(\{u,w\},z),(\{v,s\},t)\Big\},col\Big)$ for these different notations.

Definition 61 (Underlying 2-break scenario of an $\mathcal{O}$-scenario)**.** *Take an $\mathcal{O}$-scenario ρ for a labeled graph (G,λ) and omit the $\mathcal{O}$-changes to obtain a sequence of $\mathcal{O}$-breaks $\Big((\tau_1,\chi_1),\ldots,(\tau_m,\chi_m)\Big)$. The sequence of 2-breaks $(\tau_1,\ldots,\tau_m)$ is a 2-break scenario for G, we call it the* underlying *2-break scenario of ρ.*

In what follows, the scenario graph and the scenario matching (see Definition 43 and Definition 44) of a parsimonious $\mathcal{O}$-scenario for a circle will mean the scenario graph and the scenario matching of its underlying 2-break scenario.

4.3 The φ-cost of an $\mathcal{O}$-scenario

4.3.1 Optimization Problems

For an arbitrary cost function φ on a set of valid operations $\mathcal{O}$ we introduce a couple of families of optimization problems for the $\mathcal{O}$-scenarios.

Definition 62 (The φ-cost of an $\mathcal{O}$-scenario)**.** *Take a cost function $\varphi : \mathcal{O} \to \mathbb{R}_+$. The φ-cost $\varphi(\rho)$ of an $\mathcal{O}$-scenario ρ is the sum of the φ-costs of its constituent operations.*

Definition 63 (The MCS_φ-cost of a labeled graph)**.** *The MCS_φ-cost of a labeled graph (G, λ), denoted by $MCS_\varphi(G, \lambda)$, is the minimum φ-cost of an $\mathcal{O}$-scenario for (G, λ), if such an $\mathcal{O}$-scenario exists, and ∞ otherwise. A φ-MCS $\mathcal{O}$-scenario is an $\mathcal{O}$-scenario with φ-cost equal to $MCS_\varphi(G, \lambda)$.*

Definition 64 (The MCPS_φ-cost of a labeled graph)**.** *The $MCPS_\varphi$-cost of a labeled graph (G, λ), denoted by $MCPS_\varphi(G, \lambda)$, is the minimum φ-cost of a parsimonious $\mathcal{O}$-scenario for (G, λ), if such an $\mathcal{O}$-scenario exists, and ∞ otherwise. A φ-MCPS $\mathcal{O}$-scenario is a parsimonious $\mathcal{O}$-scenario with φ-cost equal to $MCPS_\varphi(G, \lambda)$.*

Problem 9 (φ-Minimum Cost Scenario or φ-MCS)**.**

$$\begin{aligned} \mathit{INPUT}&: \text{A labeled graph } (G, \lambda). \\ \mathit{OUTPUT}&: MCS_\varphi(G, \lambda). \end{aligned}$$

Problem 10 (Minimum Cost Scenario or MCS)**.**

$$\begin{aligned} \mathit{INPUT}&: \text{A pair } (\mathcal{O}, \varphi) \text{ and a labeled graph } (G, \lambda). \\ \mathit{OUTPUT}&: MCS_\varphi(G, \lambda). \end{aligned}$$

Problem 11 (φ-Minimum Cost Parsimonious Scenario or φ-MCPS)**.**

$$\begin{aligned} \mathit{INPUT}&: \text{A labeled graph } (G, \lambda). \\ \mathit{OUTPUT}&: MCPS_\varphi(G, \lambda). \end{aligned}$$

Problem 12 (Minimum Cost Parsimonious Scenario or MCPS)**.**

$$\begin{aligned} \mathit{INPUT}&: \text{A pair } (\mathcal{O}, \varphi) \text{ and a labeled graph } (G, \lambda). \\ \mathit{OUTPUT}&: MCPS_\varphi(G, \lambda). \end{aligned}$$

As we outline in the following subsection, some particular cases of φ-MCPS and φ-MCS have already been addressed in the literature. The MCS problem is more pertinent for the study of genome rearrangements, as the real evolutionary scenarios might be non-parsimonious, however it is also much more difficult to approach than

the MCPS problem. Not only have we shown MCS to be NP-complete even for fairly simple cost functions, but there is a general lack of tools for working with non-parsimonious 2-break scenarios.

This is why we first treat MCPS, which also serves as an upper bound for MCS. The results established in Chapter 3 concerning the parsimonious 2-break scenarios will lead to a polynomial time algorithm for the MCPS problem for a labeled circle and a labeled breakpoint graph in Chapter 5.

4.3.2 φ-MCPS and φ-MCS in the Literature

Minimum Local Parsimonious Scenario

In [95] we supposed the adjacencies of a genome A to be partitioned into spatial regions represented by different colors. We then developed a polynomial time algorithm for finding a parsimonious DCJ scenario minimizing the number of rearrangements whose breakpoints appear in different regions. The problem, as stated in [95], differs slightly from φ-MCPS, since in that study we do not have colors for the adjacencies of genome B. We can bridge this gap as follows.

There is a single vertex label a, while edge labels $\Sigma_E = \Sigma_c \cup \{t\}$ are the colors representing the different spatial regions of a genome plus an additional terminal label t. $\mathcal{O}$ contains 2-breaks on labels $\Big((\{a,a\},x),(\{a,a\},y);(\{a,a\},x),(\{a,a\},y);black\Big)$ for $x, y \in \Sigma_c$, and edge-label changes $\Big((\{a,a\},x);(\{a,a\},t);black\Big)$ for $x \in \Sigma_c$. The φ_c-cost of a 2-break on labels in $\mathcal{O}$ is 0 if the labels of the edge are equal and 1 otherwise. The φ_c-cost of an edge-label change is 0.

In [95] we presented an $O(n^3)$ time algorithm for the φ_c-MCPS problem on a labeled circle with the gray edges labeled with t, while in [91] we established that the φ_c-MCS problem is NP-hard. In the same paper we proposed an algorithm for the φ_c-MCS problem on a breakpoint graph that is exponential in the number of colors and not in the number of vertices. In [90] we used the same $\mathcal{O}$ and an arbitrary symmetric function $\Phi : \Sigma_E^2 \to \mathbb{R}_+$, for which we defined $\varphi_f\Big((\{a,a\},x),(\{a,a\},y);(\{a,a\},x),(\{a,a\},y);black\Big) = \Phi(x,y)$. This drastically enhanced the model introduced in [95] as now a rearrangement whose breakpoints appear in the same region could have non-zero cost. In [90] we described an $O(n^4)$ time algorithm solving φ_f-MCPS on a labeled circle on n vertices.

Sorting by wDCJs and Indels in Intergenes

Bulteau, Fertin, and Tannier [27] introduced a problem where genome adjacencies are labeled with their genetic length (number of nucleotides). A *wDCJ* is a DCJ that preserves the sum of the genetic lengths of the adjacencies and an *indel* δ

increases or decreases the genetic length of an adjacency by δ. The cost of a wDCJ is 0 and the cost of an indel δ is $|\delta|$. A scenario of wDCJs and indels for (G, λ) is said to be *valid* if its wDCJ-length is $d_{2b}(G)$. The paper treats a problem of finding a minimum cost scenario among the valid ones and presents an $O(n \log n)$ algorithm for co-tailed single-copy genomes with n genes.

Translating this into our formalism yields the following φ-MCPS problem. Edge labels are the natural numbers and there is a single vertex label that we denote by a. $\mathcal{O}$ contains 2-breaks on labels $\Big(\big(\{a,a\},w_1\big),\big(\{a,a\},w_2\big);\big(\{a,a\},w_3\big),\big(\{a,a\},w_4\big);col\Big)$ with $w_i \in \mathbb{N}$ satisfying $w_1 + w_2 = w_3 + w_4$. $\mathcal{O}$ also contains edge-label changes $\Big(\big(\{a,a\},w_1\big);\big(\{a,a\},w_2\big);col\Big)$ with $w_i \in \mathbb{N}$.

For this $\mathcal{O}$ we have that $d_{\mathcal{O}b}(G, \lambda) = d_{2b}(G)$ for any labeled graph (G, λ), since G can be first transformed into a terminal graph using any parsimonious $\mathcal{O}$-scenario and then its labels can be adjusted. The φ_l-cost of a 2-break on labels is 0 and the φ_l-cost of an edge-label change $\Big(\big(\{a,a\},w_1\big);\big(\{a,a\},w_2\big);black\Big)$ is $|w_1 - w_2|$.

For φ_l-MCPS Bulteau, Fertin, and Tannier presented an $O(r \log r)$ time algorithm on a labeled circle with r vertices. Combining this algorithm with our results from Section 5.5 provides a polynomial time algorithm for φ_l-MCPS on a labeled breakpoint graph, while the ILP defined in Section 5.2 solves φ_l-MCPS on any labeled graph.

wDCJ-dist

Fertin, Jean, and Tannier [47] treated a problem wDCJ-DIST where wDCJs without indels are allowed, and the sums of the genetic lengths of the adjacencies of two genomes are equal.

In this case we keep the same Σ_E, Σ_V and $\mathcal{O}$ as in Example 4.3.2 except that the edge-label changes are excluded from $\mathcal{O}$. A labeled graph is said to be *balanced* if the sums of the labels of black and gray edges are equal. wDCJ-DIST is the problem of finding $d_{\mathcal{O}b}(G, \lambda)$ for a balanced labeled circular graph. The authors showed that wDCJ-DIST is strongly NP-complete. However, they also proved that $d_{\mathcal{O}b}(G, \lambda) = d_{2b}(G)$ for a balanced labeled circular graph whose connected components are balanced labeled circles.

Rearrangement Scenarios that Preserve Common Intervals

Another biological constraint that has been studied for genome rearrangements is that of *conserved gene clusters*, or *preserved common intervals* [59, 12, 56]. An *interval* or *gene cluster* of a genome is a set of consecutive genes on one of its chromosomes. Gene clusters shared between genomes, also known as *common intervals*, were observed to contain functionally associated proteins [96]. This moti-

vated a search for rearrangement scenarios that preserve common intervals. Bérard, Chateau, Chauve, Paul, and Tannier [12] studied the problem of finding such a preserving DCJ scenario of the minimum length. In this subsection we introduce the $\mathcal{F}$-PERFECT DCJ problem treated in [12], and explain why it cannot be immediately interpreted as a φ-MCPS problem. We then propose a modification of $\mathcal{F}$-PERFECT DCJ that resolves this conflict. At this point our goal is to demonstrate the versatility of our method, thus further studies should be undertaken to see how these variants of $\mathcal{F}$-PERFECT DCJ compare to each other.

Take a subset of genes I. An adjacency is a *border adjacency* of I if exactly one of its gene extremities is an extremity of a gene in I. An interval of a genome, by definition, has zero or two border adjacencies. Take a genome with at most two border adjacencies of I. A DCJ transforming this genome *preserves* I if the resulting genome also contains at most two border adjacencies of I. According to this definition a preserving DCJ might excise a circular chromosome out of an interval I thus leading to a situation where I is no longer a single interval in the obtained genome. Two sets *overlap* if their intersection is non-empty and properly included in both of the sets. A family $\mathcal{F}$ of sets is *nested*, if no two sets in $\mathcal{F}$ overlap. Take two single-copy genomes A and B having n genes together with a family $\mathcal{F}$ of common intervals. A DCJ scenario transforming A into B is *$\mathcal{F}$-perfect* if its DCJs preserve the intervals in $\mathcal{F}$. The authors in [12] pose the $\mathcal{F}$-PERFECT DCJ problem of finding a minimum length $\mathcal{F}$-perfect DCJ scenario. They not only establish that the $\mathcal{F}$-PERFECT DCJ problem is NP-hard in general, but also show that it can be solved in $O(n^2)$ time for a nested $\mathcal{F}$. In this case the minimum length of a $\mathcal{F}$-perfect DCJ scenario is actually equal to $d_{aDCJ}(A, B)$.

Take a DCJ $\delta = \Big\{\{a, b\}, \{c, d\}\Big\} \to \Big\{\{a, c\}, \{b, d\}\Big\}$ and an interval I of a genome A. If the number of border adjacencies among $\Big\{\{a, b\}, \{c, d\}\Big\}$ and $\Big\{\{a, c\}, \{b, d\}\Big\}$ is equal, then δ is preserving. However if these numbers are not equal, then from δ and I alone we cannot decide whether it is preserving or not, as this now depends on A too. As a workaround we define a DCJ to *strongly preserve* a subset of genes if the resulting genome preserves the number of border adjacencies. Now no information on A is necessary to decide whether a DCJ is strongly preserving for I or not. Define a common interval for two genomes to be *strongly common* if its number of border adjacencies in both genomes is equal. If the genomes contain only linear chromosomes, which is the case for most of eukaryotic genomes, then their common interval is necessarily strongly common as it contains exactly two border adjacencies in both genomes. Take a family $\mathcal{F}$ of strongly common intervals of A and B. A DCJ scenario is *strongly $\mathcal{F}$-perfect* if its DCJs are *strongly preserving* for the intervals in $\mathcal{F}$. We pose the STRICTLY $\mathcal{F}$-PERFECT DCJ problem of finding a minimum length strongly $\mathcal{F}$-perfect DCJ scenario transforming A into B.

Take two single-copy genomes A and B with genes $\{1, \ldots, n\}$ and a family $\mathcal{F}$ of their strongly common intervals. Take a single edge label t and vertex labels

$\Sigma_V = \{1_t, \ldots, n_t\} \cup \{1_h, \ldots, n_h\}$. Construct $\mathcal{O}$ containing every 2-break on labels $\Big((\{a,b\},t),(\{c,d\},t);(\{a,c\},t),(\{b,d\},t);black\Big)$ with $a, b, c, d \in E_V$. Its φ_p-cost is 0 if the number of border adjacencies in $\Big\{\{a,b\},\{c,d\}\Big\}$ and $\Big\{\{a,c\},\{b,d\}\Big\}$ is equal for every strongly common interval in $\mathcal{F}$, it is 1 otherwise. Label the colored edges of the genome breakpoint graph $G(A,B)$ with t and its vertices with themselves to obtain a labeled graph $(G(A,B),\lambda)$. $\text{MCPS}_{\varphi_p}(G(A,B),\lambda)$ provides us with the minimum number of non-strongly preserving DCJs in a DCJ scenario of length $d_{aDCJ}(A,B)$. Using the algorithm presented in Chapter 5 we can compute it in $O(n^4)$ time.

This approach allows for a great flexibility as we can choose arbitrary costs for 2-breaks on labels. For example, define breaking costs for the intervals in $\mathcal{F}$ and define the cost of a 2-break on labels to be the sum of the breaking costs of all the intervals that it breaks. A φ-MCPS problem thus defined allows us to find a DCJ scenario of length $d_{aDCJ}(A,B)$ that best preserves a family of common intervals in $O(n^4)$ time.

Cost Constrained Transposition Decompositions of a Permutation

Farnoud and Milenkovic [41] fixed an arbitrary cost function φ on transpositions and defined the cost of a transposition decomposition of a permutation to be the sum of the costs of its transpositions. They proposed an $O(n^4)$ time algorithm for the problem of finding a minimum cost decomposition among the Minimum Length Transposition Decompositions of a permutation $\sigma \in \mathbb{S}_n$, the MIN-cost-MLTD problem. The MIN-cost-TD problem was also discussed. Farnoud and Milenkovic showed that for an arbitrary cost function MIN-cost-MLTD is a 4-approximation of MIN-cost-TD. In addition to that, Farnoud, Milenkovic, Puleo, and Su [42] conjectured that there exists a MIN-cost-TD of length $O(n^2)$.

There is a single edge label x, while vertex labels are $\Sigma_V = V_t \cup V_h$, with $V_t = \{1_t, \ldots, n_t\}$ and $V_h = \{1_h, \ldots, n_h\}$. $\mathcal{O}$ consists of the 2-breaks on labels of the form $\Big((\{u_h, v_t\}, x), (\{w_h, s_t\}, x); (\{u_h, s_t\}, x), (\{w_h, v_t\}, x); black\Big)$, where every edge is incident to a tail and a head vertex. The φ_t-cost of such a 2-break on labels is equal to $\varphi(v,s)$.

Due to Lemma 8, an instance of the MIN-cost-MLTD problem for a permutation σ can be interpreted as an instance of the φ_t-MCPS problem for its permutation breakpoint graph $H(\sigma, id)$ with every edge labeled x and every vertex labeled with itself. Analogously an instance of the MIN-cost-TD problem can be interpreted as an instance of the φ_t-MCS problem. This way the work on MIN-cost-TD [41, 42] can guide further work on φ-MCS, and our results on φ-MCPS can be used to generalize those on MIN-cost-MLTD.

Token Swapping and Interchange Rearrangement Problem

We invite the reader to revisit Section 2.7 for more examples of the problems that can be interpreted as φ-MCPS or φ-MCS.

4.4 Change-first $\mathcal{O}$-scenario

In order to solve the MCPS problem we have to search the space of the parsimonious $\mathcal{O}$-scenarios. An $\mathcal{O}$-scenario, as defined in Section 4.2, might contain any number of $\mathcal{O}$-changes, thus the search space might be prohibitively large.

Among the problems discussed in Subsection 4.3.2 only Sorting by wDCJs and Indels in Intergenes [27] (or φ_l-MCPS in our notation) allowed for $\mathcal{O}$-changes. The authors established that there exists an MCPS_{φ_l} scenario consisting of a sequence of $\mathcal{O}$-breaks followed by a sequence of $\mathcal{O}$-changes. This observation drastically reduced the search space. Along the same lines, we introduce a notion of a *change-first* $\mathcal{O}$-scenario. It is an $\mathcal{O}$-scenario that starts with some $\mathcal{O}$-changes on the labeled edges of a labeled graph and then proceeds with $\mathcal{O}$-breaks without any more $\mathcal{O}$-changes. We would like to explore the space of the change-first $\mathcal{O}$-scenarios and not that of all the $\mathcal{O}$-scenarios. To this end we introduce a notion of a *complete* pair $(\mathcal{O}, \varphi)$, and show that in this setting for every labeled graph there exists a change-first MCPS_{φ} $\mathcal{O}$-scenario. In the following section we demonstrate that every pair $(\mathcal{O}, \varphi)$ can be transformed into a complete one in polynomial time, which allows us to concentrate solely on the change-first $\mathcal{O}$-scenarios in Chapter 5.

Definition 65 (Complete $(\mathcal{O},\varphi)$)**.** *A pair $(\mathcal{O}, \varphi)$ is* complete *if the following properties hold.*

1. *If $\mathcal{O}$ contains a pair of edge-label changes $\gamma_0 = \Big((\{a,b\}, x_0); (\{a,b\}, x_1); col\Big)$ and $\gamma_1 = \Big((\{a,b\}, x_1); (\{a,b\}, x_2); col\Big)$, then $\mathcal{O}$ also contains an edge-label change $\gamma_2 = \Big((\{a,b\}, x_0); (\{a,b\}, x_2); col\Big)$, and $\varphi(\gamma_2) \leq \varphi(\gamma_0) + \varphi(\gamma_1)$.*
2. *If $\mathcal{O}$ contains $\pi_0 = \Big(\Big\{(\{a,b\}, x), (\{c,d\}, y)\Big\}; \Big\{(\{a,c\}, z), (\{b,d\}, t)\Big\}; col\Big)$, $\gamma_0 = \Big((\{a,c\}, z); (\{a,c\}, z'); col\Big)$ and $\gamma_1 = \Big((\{b,d\}, t); (\{b,d\}, t'); col\Big)$, then $\mathcal{O}$ also contains $\pi_1 = \Big(\Big\{(\{a,b\}, x), (\{c,d\}, y)\Big\}; \Big\{(\{a,c\}, z'), (\{b,d\}, t')\Big\}; col\Big)$, for which $\varphi(\pi_1) \leq \varphi(\pi_0) + \varphi(\gamma_0) + \varphi(\gamma_1)$.*

Definition 66 (Change-first $\mathcal{O}$-scenario)**.** *An $\mathcal{O}$-scenario for a labeled graph (G, λ) is* change-first *if it is a sequence of $\mathcal{O}$-changes with at most one $\mathcal{O}$-change per labeled edge of (G, λ), followed by a sequence of $\mathcal{O}$-breaks.*

See Figure 4.2 for an example of an $\mathcal{O}$-scenario and an illustration of Lemma 25 in which we provide a canonical $\mathcal{O}$-scenario $\hat{\rho}$ preserving some of the core properties

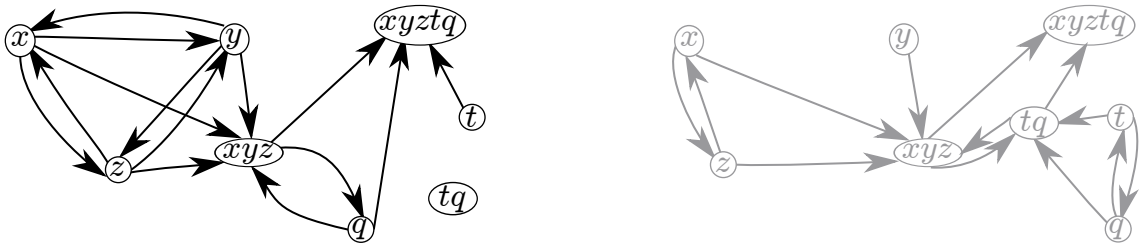


Figure 4.1: In this figure we provide an example of a pair $(\mathcal{O}, \varphi)$ and its completion $(\mathcal{O}_c, \varphi_c)$. For this we use a single vertex label a and edge labels $\{x, y, z, t, q, qt, xyz, xyztq\}$. The figure itself depicts 1-edge-colored directed multigraphs $W^{black}_{\{a,a\}}$ and $W^{gray}_{\{a,a\}}$ introduced in Lemma 26 whose directed edges define the edge-label changes in $\mathcal{O}$ as follows. An edge-label change $\gamma_1 = \Big(\big(\{a,a\},x\big);\big(\{a,a\},y\big);black\Big)$ is in $\mathcal{O}$, as a directed edge (x,y) is in $W^{black}_{\{a,a\}}$, and $\gamma_2 = \Big(\big(\{a,a\},x\big);\big(\{a,a\},q\big);black\Big)$ is not in $\mathcal{O}$, as $W^{black}_{\{a,a\}}$ does not contain a directed edge (x,q). $\mathcal{O}$ contains the 2-breaks on labels involving a single edge label. For example a 2-break on labels $\pi_1 = \Big(\Big\{\big(\{a,a\},x\big),\big(\{a,a\},x\big)\Big\};\Big\{\big(\{a,a\},x\big),\big(\{a,a\},x\big)\Big\};black\Big)$ is in $\mathcal{O}$, as it involves a single edge label x, and $\pi_2 = \Big(\Big\{\big(\{a,a\},x\big),\big(\{a,a\},x\big)\Big\};\Big\{\big(\{a,a\},y\big),\big(\{a,a\},q\big)\Big\};black\Big)$ is not in $\mathcal{O}$, as it involves 3 existing edge labels. We define the φ-cost of an edge-label change in $\mathcal{O}$ to be equal to 1 and the φ-cost of a 2-break on labels to be equal to the length of the involved edge label. For example $\varphi(\pi_1) = 1$, as the length of x is 1. We provide a short description of the completion $(\mathcal{O}_c, \varphi_c)$ of $(\mathcal{O}, \varphi)$ introduced in Section 4.5. An edge-label change γ_2 belongs to $\mathcal{O}_c$, as there is a directed path from x to q in $W^{black}_{\{a,a\}}$. The minimum length of such a path is 2, thus $\varphi_c(\gamma_2) = 2$. A 2-break on labels π_2 is in $\mathcal{O}_c$, as π_1 is in $\mathcal{O}$, and γ_1 and γ_2 are in $\mathcal{O}_c$, furthermore $\varphi_c(\pi_2)$ is equal to $\varphi(\pi_1) + \varphi_c(\gamma_1) + \varphi_c(\gamma_2) = 4$.

of an $\mathcal{O}$-scenario ρ. Lemma 25 will be used in the following chapter when solving the φ-MCPS problem for a circle.

Lemma 25. *Take a complete pair ($\mathcal{O}$,φ) and an $\mathcal{O}$-scenario ρ for a labeled circle (C,λ). There exists a change-first $\mathcal{O}$-scenario $\hat{\rho}$ satisfying the following properties:*

- $\varphi(\hat{\rho}) \leq \varphi(\rho)$,
- *The underlying 2-break scenarios of $\hat{\rho}$ and ρ are equal,*
- *If ρ does not contain an $\mathcal{O}$-change modifying the label of some colored edge, then neither does $\hat{\rho}$.*

Proof. Take $(\tau, \chi) = \Big(\Big\{\big(\{u,v\},x\big),\big(\{w,s\},y\big)\Big\} \to \Big\{\big(\{u,w\},z\big),\big(\{v,s\},t\big)\Big\}, col\Big)$, the first $\mathcal{O}$-break in ρ. Denote by $\rho_{\{u,w\}}$ the subsequence of ρ consisting of the $\mathcal{O}$-changes applying to a colored edge $(\{u,w\}, col)$. Keep only those $\mathcal{O}$-changes in $\rho_{\{u,w\}}$ that

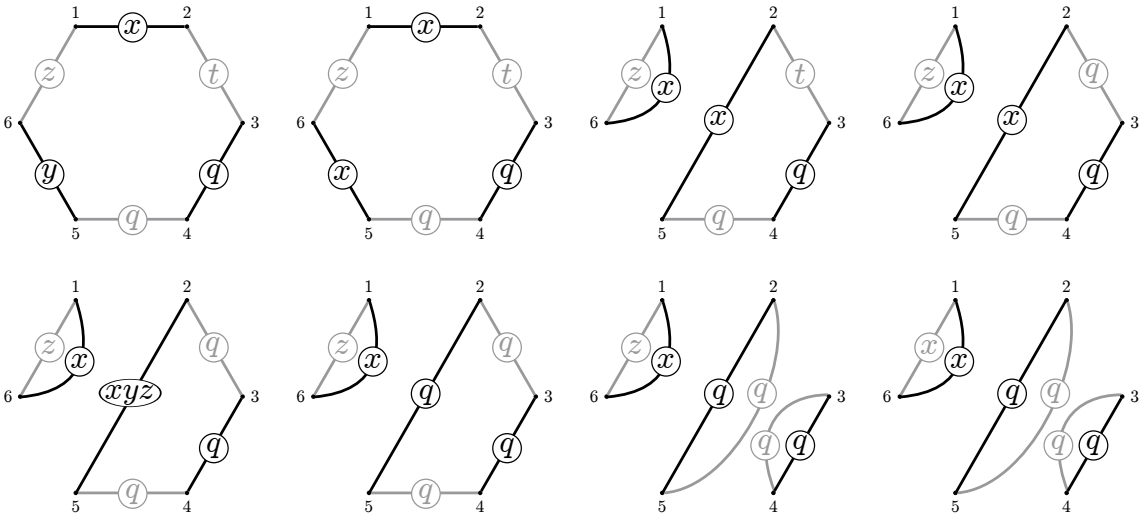


Figure 4.2: We provide an example of an $\mathcal{O}_c$-scenario for a labeled circle using a set of valid operations $\mathcal{O}_c$ introduced in Figure 4.1. $\rho = \Big($ $(\{5,6\}, black, y) \to (\{5,6\}, black, x)$, $\Big(\Big\{(\{1,2\}, x), (\{5,6\}, x)\Big\} \to \Big\{(\{1,6\}, x), (\{2,5\}, x)\Big\}, black\Big)$, $(\{2,3\}, gray, t) \to (\{2,3\}, gray, q)$, $(\{2,5\}, black, x) \to (\{2,5\}, black, xyz)$, $(\{2,5\}, black, xyz) \to (\{2,5\}, black, q)$, $\Big(\Big\{(\{2,3\}, q), (\{4,5\}, q)\Big\} \to \Big\{(\{2,5\}, q), (\{3,4\}, q)\Big\}, gray\Big)$, $(\{1,6\}, gray, z) \to (\{1,6\}, gray, x)\Big)$. A possible change-first $\mathcal{O}_c$-scenario obtained from ρ in Lemma 25 is $\hat{\rho} = \Big($ $(\{5,6\}, black, y) \to (\{5,6\}, black, x)$, $(\{2,3\}, gray, t) \to (\{2,3\}, gray, q)$, $(\{1,6\}, gray, z) \to (\{1,6\}, gray, x)\Big)$, $\Big(\Big\{(\{1,2\}, x), (\{5,6\}, x)\Big\} \to \Big\{(\{1,6\}, x), (\{2,5\}, q)\Big\}, black\Big)$, $\Big(\Big\{(\{2,3\}, q), (\{4,5\}, q)\Big\} \to \Big\{(\{2,5\}, q), (\{3,4\}, q)\Big\}, gray\Big)\Big)$. It is not unique, as the $\mathcal{O}_c$-changes in the beginning of the change-first scenario can be performed in any order.

follow (τ, χ) but precede the following $\mathcal{O}$-break replacing $(\{u, w\}, col)$. If $\rho_{\{u,w\}}$ is not empty, then denote by z' the label provided by the last $\mathcal{O}$-change in $\rho_{\{u,w\}}$, otherwise denote $z' = z$. Due to Corollary 2 in Section 2.2, every prefix of ρ transforms (C, λ) into a union of vertex disjoint circles, and, by definition, a circle contains at most one copy of every colored edge. This means that all the $\mathcal{O}$-changes in $\rho_{\{u,w\}}$ apply to the same colored edge. Analogously, construct the subsequence $\rho_{\{v,s\}}$ of the $\mathcal{O}$-changes in ρ transforming the label of $(\{v, s\}, col)$ from t to t'. Due to the completeness of $(\mathcal{O}, \varphi)$, $\Big(\Big\{(\{u, v\}, x), (\{w, s\}, y)\Big\} \to \Big\{(\{u, w\}, z'), (\{v, s\}, t')\Big\}, col\Big)$ is an $\mathcal{O}$-break. Transform ρ by deleting $\rho_{\{u,w\}}$ and $\rho_{\{v,s\}}$, and replacing (τ, χ) with $\Big(\Big\{(\{u, v\}, x), (\{w, s\}, y)\Big\} \to \Big\{(\{u, w\}, z'), (\{v, s\}, t')\Big\}, col\Big)$, to obtain an $\mathcal{O}$-scenario of smaller or equal φ-cost. The underlying 2-break scenario of ρ remains unchanged. Proceed with the rest of $\mathcal{O}$-breaks in ρ.

Now take a labeled edge $(\{u, v\}, col, x)$ of (C, λ). As above, construct a subse-

quence $\rho_{\{u,v\}}$ containing the $\mathcal{O}$-changes in ρ transforming the label of $(\{u,v\}, col)$ from x to x'. If $\rho_{\{u,v\}}$ is not empty, then transform ρ by adding $(\{u,v\}, col, x) \to (\{u,v\}, col, x')$ to its beginning and deleting $\rho_{\{u,w\}}$ to obtain an $\mathcal{O}$-scenario of lower or equal φ-cost. The underlying 2-break scenario of ρ remains unchanged. Proceed with the rest of labeled edges of (C, λ) to obtain a change-first $\mathcal{O}$-scenario. □

Corollary 8. *Take a complete pair $(\mathcal{O}, \varphi)$ and a labeled circle (C, λ). Either there exists a change-first φ-MCPS $\mathcal{O}$-scenario for (C, λ), or $MCPS_{\varphi}(C, \lambda) = \infty$.*

4.5 Completion of $(\mathcal{O}, \varphi)$

In this section we transform a pair $(\mathcal{O}, \varphi)$ into a complete pair $(\mathcal{O}_c, \varphi_c)$, while ensuring that MCPS_{φ} and MCPS_{φ_c} costs for every labeled graph are equal.

Definition 67 (Completion of $(\mathcal{O}, \varphi)$)**.** *Construct the* completion *$(\mathcal{O}_c,\varphi_c)$ of $(\mathcal{O},\varphi)$ as follows. To begin with, for every sequence of edge-label changes from $\mathcal{O}$ of the form*

$$\Big(\big((\{a,b\}, x_0); (\{a,b\}, x_1); col\big), \big((\{a,b\}, x_1); (\{a,b\}, x_2); col\big), \ldots, \big((\{a,b\}, x_{m-1}); (\{a,b\}, x_m); col\big)\Big)$$

add an edge-label change $\big((\{a,b\}, x_0); (\{a,b\}, x_m); col\big)$ to $\mathcal{O}_c$. A number of different sequences might result in the same edge-label change in $\mathcal{O}_c$, define its φ_c-cost to be the minimum sum of the φ-costs of the edge-label changes in such a sequence. For every 2-break on labels $\Big(\big\{(\{a,b\}, x), (\{c,d\}, y)\big\}; \big\{(\{a,c\}, z), (\{b,d\}, t)\big\}; col\Big)$ in $\mathcal{O}$, and a pair of edge-label changes $\big((\{a,c\}, z); (\{a,c\}, z'); col\big)$ and $\big((\{b,d\}, t); (\{b,d\}, t'); col\big)$ in $\mathcal{O}_c$, add a 2-break on labels $\Big(\big\{(\{a,b\}, x), (\{c,d\}, y)\big\}; \big\{(\{a,c\}, z'), (\{b,d\}, t')\big\}; col\Big)$ to $\mathcal{O}_c$. A number of different combinations of a 2-break on labels in $\mathcal{O}$ and edge-label changes in $\mathcal{O}_c$ might result in the same 2-break on labels in $\mathcal{O}_c$, define its φ_c-cost as the minimum sum of the φ-cost of a 2-break on labels and the φ_c-costs of the edge-label changes invoking it.

Observation 20. *$(\mathcal{O}_c, \varphi_c)$ is complete by construction.*

See Figure 4.1 for an example of the completion of $(\mathcal{O}, \varphi)$ and an illustration for Lemma 26.

Lemma 26. *The completion $(\mathcal{O}_c,\varphi_c)$ of $(\mathcal{O},\varphi)$ can be constructed in $O(|E_V|^2|\Sigma_E|^3 + |\Sigma_E|^2|\mathcal{O}|)$ time, which is $O(|\Sigma_E|^6|E_V|^4)$ in the worst case.*

Proof. For a color $col \in$ {black,gray} and a pair of vertex labels $\{a,b\}$, define an edge-weighted 1-edge-colored directed multigraph $W^{col}_{\{a,b\}}$ with a vertex set Σ_E. For

an edge-label change $\Big((\{a,b\},x);(\{a,b\},y);col\Big)$ in $\mathcal{O}$, add a directed edge (x,y) in $W^{col}_{\{a,b\}}$ of weight $\varphi\Big((\{a,b\},x);(\{a,b\},y);col\Big)$. For a pair of edge labels $\{x,y\}$, an edge-label change $\Big((\{a,b\},x);(\{a,b\},y);col\Big)$ is in $\mathcal{O}_c$ if and only if a path from x to y exists in $W^{col}_{\{a,b\}}$. Further, its φ-cost is equal to the minimum weight of such a path. Johnson's algorithm [63] can be used to find the minimum weight paths between all the pairs of vertices in $W^{col}_{\{a,b\}}$ in $O(|\Sigma_E|^3)$ time. This operation has to be repeated for the $|E_V|^2$ pairs of vertex labels and two different colors, thus φ_c-costs of the edge-label changes in $\mathcal{O}_c$ can be computed in $O(|E_V|^2|\Sigma_E|^3)$ time.

Take a 2-break on labels $\Big(\Big\{(\{a,b\},x),(\{c,d\},y)\Big\};\Big\{(\{a,c\},z),(\{b,d\},t)\Big\};col\Big)$ in $\mathcal{O}$ and a pair of edge labels $\{z',t'\}$. If edge-label changes $\Big((\{a,c\},z);(\{a,c\},z');col\Big)$ and $\Big((\{b,d\},t);(\{b,d\},t');col\Big)$ are in $\mathcal{O}_c$, then it also contains a 2-break on labels $\Big(\Big\{(\{a,b\},x),(\{c,d\},y)\Big\};\Big\{(\{a,c\},z'),(\{b,d\},t')\Big\};col\Big)$. One can iterate through 2-breaks on labels in $\mathcal{O}$ and the pairs of edge labels in $O(|\mathcal{O}||\Sigma_E|^2)$ time. If one can check in constant time whether an edge-label change is present in $\mathcal{O}_c$, then the 2-breaks on labels in $\mathcal{O}_c$ and their φ_c-costs can also be obtained in $O(|\mathcal{O}||\Sigma_E|^2)$ time. The size of $\mathcal{O}$ is $O(|\Sigma_V|^4|\Sigma_E|^4)$ in the worst case, which means that $O(|E_V|^2|\Sigma_E^3| + |\Sigma_E|^2|\mathcal{O}|)$ is $O(|\Sigma_E|^6|E_V|^4)$. □

Lemma 27. *$MCPS_{\varphi}(G,\lambda) = MCPS_{\varphi_c}(G,\lambda)$ for a labeled graph (G,λ).*

Proof. $\mathrm{MCPS}_{\varphi_c}(G,\lambda) \leq \mathrm{MCPS}_{\varphi}(G,\lambda)$ as every $\mathcal{O}$-scenario for (G,λ) is also an $\mathcal{O}_c$-scenario for (G,λ). This means that if $\mathrm{MCPS}_{\varphi_c}(G,\lambda) = \infty$, then $\mathrm{MCPS}_{\varphi}(G,\lambda)$ is also ∞. Suppose that a parsimonious $\mathcal{O}_c$-scenario ρ for (G,λ) exists. By construction, every $\mathcal{O}_c$-break in ρ can be replaced with an $\mathcal{O}$-break followed by a pair of $\mathcal{O}_c$-changes, and every $\mathcal{O}_c$-change can be replaced by a sequence of $\mathcal{O}$-changes to obtain an $\mathcal{O}$-scenario with φ-cost equal to φ_c-cost of ρ. □

Corollary 8 and Lemma 27 ensure that for every labeled graph there exists a change-first $\mathcal{O}_c$-scenario of φ_c-cost equal the MCPS_{φ}-cost of that labeled graph.

4.6 Conclusion

We have introduced a framework for cost constraining 2-breaks and demonstrated that it generalizes previous work on cost constrained DCJ rearrangements and cost constrained transpositions.

We have shown that when dealing with the Minimum Cost Parsimonious Scenario problem it is enough to explore the space of the change-first $\mathcal{O}$-scenarios. These are the $\mathcal{O}$-scenarios in which the $\mathcal{O}$-changes precede the $\mathcal{O}$-breaks. However for this to work we first need to construct the completion $(\mathcal{O}_c,\varphi_c)$, which, as presented in Lemma 26, is a time consuming task. In practice this might prompt us

to either start with a complete pair $(\mathcal{O}, \varphi)$ from the beginning, or use some additional knowledge about a particular pair $(\mathcal{O}, \varphi)$ to devise a faster algorithm for constructing its completion.

Chapter 5

Minimum Cost Parsimonious Scenario

5.1 Introduction

In this chapter we treat the MCPS problem. The central result is a polynomial time dynamic programming algorithm for MCPS on a labeled circle for a complete pair $(\mathcal{O}, \varphi)$. The MCPS problem is NP-hard in general due to the NP-hardness of finding a Maximum Alternating Edge-disjoint Cycle Decomposition of a graph [19], nevertheless we demonstrate that our algorithm for a labeled circle can be used as a subroutine to solve MCPS for any labeled graph. This results in a proof that MCPS remains polynomial time solvable for a labeled breakpoint graph.

We build on the work presented in Chapter 3. The *Labeled Eulerian decomposition*, the *labeled trajectory graph*, the *labeled circle of a simple cycle*, and the *labeled sub-circle* are natural generalizations of the structures previously used to study parsimonious 2-break scenarios. We show that the proofs from Chapter 3 can be adapted to this labeled setting and prove the following. First we show that the MCPS_φ-cost of a labeled graph is equal to the minimum of the MCPS_φ-costs of its labeled Maximum Alternating Edge-disjoint Cycle Decompositions. We proceed by establishing that the MCPS_φ-cost of a labeled simple cycle is equal to the minimum of the MCPS_φ-costs of its labeled circles. Finally we show that a parsimonious $\mathcal{O}$-scenario for a circle can be partitioned into parsimonious $\mathcal{O}$-scenarios for its labeled sub-circles, which leads to a dynamic programming algorithm for MCPS. Its worst case time complexity is $O(n^4L^4)$, where n is the number of vertices in a circle and L is the number of edge labels used in $\mathcal{O}$.

Our algorithm is general in that it works with an arbitrary cost function. If a particular cost function φ is chosen, then its combinatorial properties might be exploited to come up with a faster algorithm for a particular φ-MCPS problem. No-

tably this is a case for the problems introduced in Section 4.3.2, namely Sorting by wDCJs and Indels in Intergenes [27] and Minimum Local Parsimonious Scenario [95]. For these problems $O(n\log(n))$ and $O(n^3)$ time algorithms exist for a labeled circle on n vertices. In Section 5.5 we explain how such a preexisting $O(n^t)$ time algorithm with $t \geq 1$ for φ-MCPS on a labeled circle can be used as a subroutine to obtain an $O(n^{t+1})$ time algorithm for φ-MCPS on a labeled breakpoint graph.

5.2 MCPS for a Graph

5.2.1 Introduction

Here we build on Theorem 4 stating that a parsimonious 2-break scenario for a graph can be partitioned into parsimonious 2-break scenarios for the subgraphs in its Maximum Alternating Edge-disjoint Cycle Decomposition. We adjust its proof to show that the same holds for a labeled graph and its parsimonious $\mathcal{O}$-scenario. This allows us to prove that the MCPS$_\varphi$-cost of a labeled graph is equal to the minimum of the MCPS$_\varphi$-costs of its *labeled MAECDs*. We proceed by proposing a straightforward ILP for computing the MCPS$_\varphi$-cost of a labeled graph once the MCPS$_\varphi$-costs of its labeled simple cycles are known.

5.2.2 The MCPS$_\varphi$-cost of a Labeled Graph is Equal to the Minimum MCPS$_\varphi$-cost of its labeled MAECD

Definition 68 (Labeled subgraph)**.** *(H, λ_H) is a* labeled subgraph *of a labeled graph (G, λ), if H is a subgraph of G and λ_H coincides with λ on the vertices and the colored edges of H.*

Definition 69 (Labeled Eulerian decomposition (ED) and its MCPS$_\varphi$-cost)**.** *$\mathcal{H} = \{(H_1, \lambda_{H_1}), \ldots, (H_k, \lambda_{H_k})\}$ is a* labeled *Eulerian decomposition of a labeled graph (G, λ), if the following properties hold:*

- *$\{H_1, \ldots, H_k\}$ is an ED of G.*
- *Every element in $\mathcal{H}$ is a labeled subgraph of (G, λ).*
- *For every labeled edge its multiplicity in (G, λ) is equal to the sum of multiplicities in $\{(H_1, \lambda_{H_1}), \ldots, (H_k, \lambda_{H_1})\}$.*

The MCPS$_\varphi$-cost of $\mathcal{H}$ *is equal to the sum of the MCPS$_\varphi$-costs of its labeled subgraphs.*

Observation 21. *Take a labeled Eulerian decomposition of a labeled graph (G, λ) and a set of $\mathcal{O}$-scenarios for its labeled subgraphs. A shuffle (see Definition 25 in Section 3.2) of these $\mathcal{O}$-scenarios is an $\mathcal{O}$-scenario for (G, λ). If the $\mathcal{O}$-scenarios for the subgraphs are parsimonious, then the obtained $\mathcal{O}$-scenario for (G, λ) is also parsimonious.*

Lemma 28. *Take a parsimonious $\mathcal{O}$-scenario ρ for a labeled graph (G, λ). There exists a labeled MAECD of (G, λ), such that ρ is a shuffle of the parsimonious $\mathcal{O}$-scenarios for its labeled subgraphs.*

Proof. The proof is, up to minor modifications, analogous to the proof of Theorem 4. These modifications are the following. The labels of the directed 2-edge-colored edge-labeled trajectory graph $\mathcal{D}((G, \lambda), \rho)$ are of the form $(\{u, v\}, x)$ for a pair of vertices $\{u, v\}$ of G and an edge label x. If the l-th operation of ρ is an $\mathcal{O}$-change $(\{u, v\}, col, x) \to (\{u, v\}, col, y)$, then $\mathcal{D}((G, \lambda), \rho_l)$ is obtained from $\mathcal{D}((G, \lambda), \rho_{l-1})$ by choosing a sink edge of $\mathcal{D}((G, \lambda), \rho_{l-1})$ of color *col* labeled $(\{u, v\}, x)$, and adding an edge of color *col* labeled $(\{u, v\}, y)$ between its sink vertex and the newly added one. This transformation does not modify the number of the connected components of $\mathcal{D}((G, \lambda), \rho_{l-1})$. Non-source and non-sink vertices of a connected component of $\mathcal{D}((G, \lambda), \rho)$ correspond to a subsequence of ρ that is a parsimonious $\mathcal{O}$-scenario for a labeled subgraph of (G, λ). These labeled subgraphs corresponding to the connected components of $\mathcal{D}((G, \lambda), \rho)$ form a labeled MAECD of (G, λ). □

Corollary 9. *A parsimonious $\mathcal{O}$-scenario for a labeled graph having two connected components can be partitioned into two sub-sequences that are parsimonious $\mathcal{O}$-scenarios for these components.*

Theorem 13. *The $MCPS_\varphi$-cost of a labeled graph is equal to the minimum $MCPS_\varphi$-cost of its labeled MAECD.*

Proof. For a labeled graph (G, λ), take its minimum MCPS_φ-cost labeled MAECD $\mathcal{H} = \{(H_1, \lambda_{H_1}), \ldots, (H_k, \lambda_{H_k})\}$. If the MCPS_φ-cost of $\mathcal{H}$ is not equal to ∞, then there exist parsimonious $\mathcal{O}$-scenarios for its labeled subgraphs of the φ-costs equal to the MCPS_φ-costs of those labeled subgraphs. Due to Observation 21, by performing these parsimonious $\mathcal{O}$-scenarios one after another we obtain a parsimonious $\mathcal{O}$-scenario ρ for (G, λ). Its φ-cost is equal to the MCPS_φ-cost of the labeled MAECD under question, establishing that $\mathrm{MCPS}_\varphi(G, \lambda)$ is less than or equal to the minimum MCPS_φ-cost of a labeled MAECD of (G, λ).

If $\mathrm{MCPS}_\varphi(G, \lambda) \neq \infty$, then take an MCPS_φ $\mathcal{O}$-scenario for (G, λ). Due to Lemma 28, there exists a labeled MAECD of (G, λ), such that ρ is a shuffle of the parsimonious $\mathcal{O}$-scenarios for its labeled subgraphs. The φ-cost of ρ is equal to the sum of the φ-costs of those parsimonious $\mathcal{O}$-scenarios, establishing that $\mathrm{MCPS}_\varphi(G, \lambda)$ is greater than or equal to the MCPS_φ-cost of a labeled MAECD of (G, λ). □

5.2.3 An ILP for φ-MCPS

In the following sections we will show how to compute the MCPS_φ-costs of labeled simple cycles of a labeled graph. Once these costs are known, Theorem 13 leads to a straightforward ILP computing the MCPS_φ-cost of (G, λ).

Denote by $\mathcal{S}$ a set of the simple cycles of G. For every simple cycle S in $\mathcal{S}$ assign a variable x_S, with x_S equal to 1 if S belongs to a cycle packing and 0 otherwise. Denote by E a set of the colored edges present in G and by $mult(G, e)$ the multiplicity of a colored edge e in G. Start by computing the size of an MAECD of G by packing as many simple cycles in G as possible.

$$\begin{aligned} \text{Maximize } & \textstyle\sum_{S\in\mathcal{S}} x_S \\ \text{Subject to } & \textstyle\sum_{S:e\in\mathcal{S}} x_S mult(S,e) \leq mult(G,e) \text{ for } e \in E, \\ & \text{and } x_S \in \{0,1\} \text{ for } S \in \mathcal{S}. \end{aligned}$$

Now denote by $\mathcal{S}^\lambda$ a set of the labeled simple cycles that are labeled subgraphs of (G, λ), by E^λ a set of the labeled edges present in (G, λ), by $mult^\lambda((G, \lambda), e)$ the multiplicity of a labeled edge e in (G, λ), and by $c(G)$ the size of an MAECD of G. The following ILP computes $\text{MCPS}_\varphi(G, \lambda)$.

$$\begin{aligned} \text{Minimize } & \textstyle\sum_{(S,\lambda_S)\in\mathcal{S}^\lambda} x_{(S,\lambda_S)} \text{MCPS}_\varphi(S,\lambda_S) \\ \text{Subject to } & \textstyle\sum_{(S,\lambda_S):e\in\mathcal{S}^\lambda} x_{(S,\lambda_S)} mult^\lambda((S,\lambda_S),e) \leq mult^\lambda((G,\lambda),e) \text{ for } e \in E^\lambda, \\ & \textstyle\sum_{(S,\lambda_S)\in\mathcal{S}^\lambda} x_{(S,\lambda_S)} = c(G) \text{ and } x_{(S,\lambda_S)} \in \{0,1\} \text{ for } (S,\lambda_S) \in \mathcal{S}^\lambda. \end{aligned}$$

5.2.4 φ-MCPS for Isomorphic Labeled Graphs

In this subsection we introduce a notion of *isomorphic labeled graphs* and establish in Lemma 29 that their MCPS_φ-costs are equal. It will be used in the following section where we show that for a *labeled circle of a labeled simple cycle* there exists an isomorphic *labeled u-circle*. Similarly, in Theorem 6 we have shown that for a circle of a simple cycle there exists an equivalent u-circle.

Definition 70 (Isomorphism of labeled graphs)**.** *Two labeled graphs (G^1, λ^1) and (G^2, λ^2) on vertices V are isomorphic if there exists a function $g : V \to V$ for which the following properties hold:*

- *For every vertex v the labels of v in (G^1, λ^1) and $g(v)$ in (G^2, λ^2) are equal.*
- *For every colored edge $(\{u, v\}, col)$ the multisets of edge labels of $(\{u, v\}, col)$ in (G^1, λ^1) and of $(\{g(u), g(v)\}, col)$ in (G^2, λ^2) are equal.*

Such a function g is an isomorphism *of (G^1, λ^1) and (G^2, λ^2).*

See Figure 5.1 for an example of isomorphic and non-isomorphic labeled circles.

Lemma 29. *The MCPS$_\varphi$-costs of two isomorphic labeled graphs are equal.*

Proof. Take a pair of isomorphic labeled graphs (G^1, λ^1) and (G^2, λ^2) together with their isomorphism g and a labeled edge $(\{u, v\}, col, x)$ of (G^1, λ^1). By definition (G^2, λ^2) contains a labeled edge $(\{g(u), g(v)\}, col, x)$ and vertex labels of u and $g(u)$, and v and $g(v)$ are equal. This means that the φ-costs of $\mathcal{O}$-changes $(\{u, v\}, col, x) \to (\{u, v\}, col, y)$ and $(\{g(u), g(v)\}, col, x) \to (\{g(u), g(v)\}, col, y)$ are also equal. In addition to that they transform (G^1, λ^1) and (G^2, λ^2) into isomorphic labeled graphs with g being their isomorphism. An analogous result can be established for an $\mathcal{O}$-change transforming (G^1, λ^1). Taken together these observations show that to a φ-MCPS $\mathcal{O}$-scenario for (G^1, λ^1) we can assign an $\mathcal{O}$-scenario for (G^2, λ^2) of equal φ-cost. We can do the same for a φ-MCPS $\mathcal{O}$-scenario for (G^2, λ^2) which finishes the proof. □

5.3 φ-MCPS for a Simple Cycle

Here we build on Theorem 6 in Section 3.4.5 stating that a 2-break scenario for a simple cycle can be interpreted as a 2-break scenario for its u-circle. We adjust its proof to show that the MCPS$_\varphi$-cost of a labeled simple cycle is equal to the minimum of the MCPS$_\varphi$-costs of its labeled u-circles.

We start by introducing the notion of a *labeled circle of a simple cycle*. Take a simple cycle S on vertices V. As defined in Section 3.4, a vertex of S is double if its black and gray degrees are equal to two, it is single otherwise. $\hat{V}$ is a set of vertices in which every double vertex v in S replaced with new vertices v^1 and v^2. M is a function $\hat{V} \to V$ satisfying $M(v) = v$ for a single vertex of S, and $M(v^1) = M(v^2) = v$ for a double vertex of S.

Definition 71 (Labeled circle of a simple cycle)**.** *A labeled circle $(C, \hat{\lambda})$ is a* labeled circle of a simple labeled cycle *(S, λ) if the following properties hold:*

- *C is a circle of S.*
- *The labels given by λ and $\hat{\lambda}$ coincide on the single vertices of S.*
- *For every double vertex v of S, the labels of v^1 and v^2 given by $\hat{\lambda}$ are equal to the label of v given by λ.*
- *Take a labeled edge $(\{M(u), M(v)\}, col, x)$ for every labeled edge $(\{u, v\}, col, x)$ of $(C, \hat{\lambda})$. A multiset of labeled edges thus obtained is equal to the labeled edges of (S, λ).*

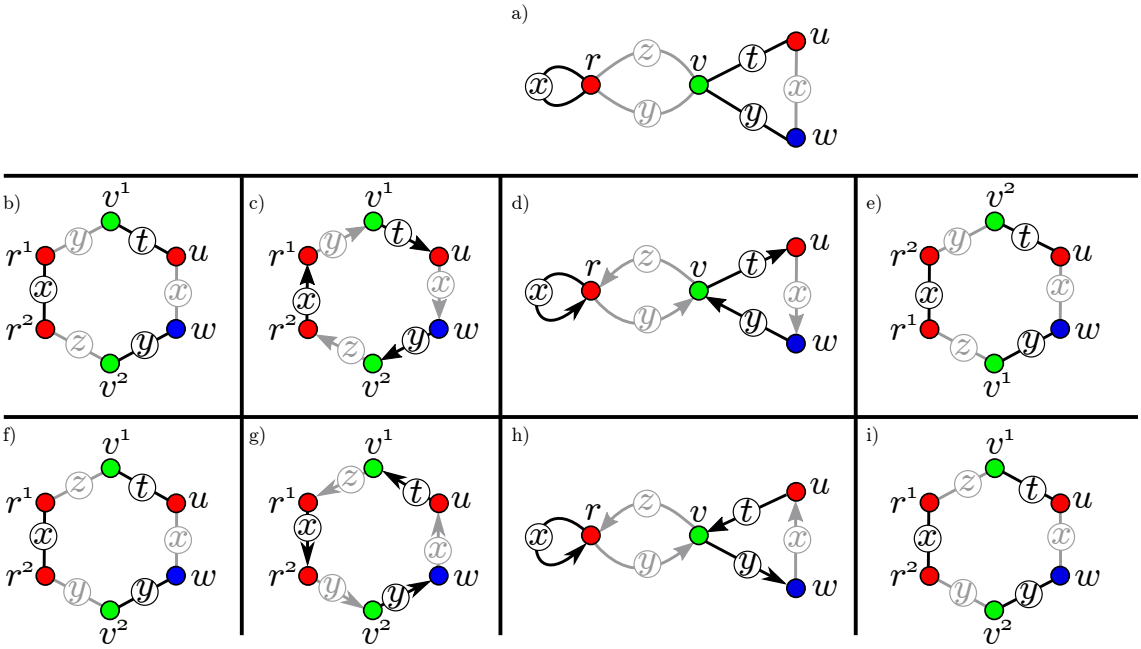


Figure 5.1: A labeled simple cycle (S, λ) is depicted in a) with vertex labels represented using different colors. In b) and f) two labeled circles of (S, λ) are presented. They are not isomorphic, as a labeled circle in b) contains a vertex incident to colored edges labeled y and t, while such a vertex does not exist in a labeled circle in f). In c) and g) possible Eulerian orientations of these labeled circles are depicted. Once pairs of vertices $\{v^1, v^2\}$ and $\{r^1, r^2\}$ are merged into v and r, these Eulerian orientations of labeled circles become Eulerian orientations of (S, λ) depicted in d) and h). The vertex sequences of their u-tours (u, w, v, r, r, v, u) and (u, v, r, r, v, w, u). The labeled u-circles obtained from these u-tours following the process detailed in Section 3.4.5 are presented in e) and i). The labeled circles presented in b) and e) are isomorphic, while the ones presented in f) and i) are even equal.

See Figure 5.1 for an example of a labeled circle, a *labeled Eulerian orientation* and its *labeled u-circle*.

Theorem 14. *Take a vertex u of a labeled simple cycle (S, λ). $MCPS_\varphi(S, \lambda)$ is equal to the minimum of the $MCPS_\varphi$-costs of the labeled u-circles of (S, λ).*

Proof. Take a labeled u-circle of (S, λ) and its $\mathcal{O}$-scenario $\hat{\rho}$. For every double vertex v in S, replace the occurrences of v^1 and v^2 in $\hat{\rho}$ with v to obtain an $\mathcal{O}$-scenario ρ for (S, λ). The φ-costs of ρ and $\hat{\rho}$ are equal as the labels of v^1, v^2 and v are equal by construction. This means that $\mathrm{MCPS}_\varphi(S, \lambda)$ is less than or equal to the minimum MCPS-cost of a labeled u-circle of (S, λ).

In Theorem 5 to a 2-break scenario for a simple cycle we assigned a 2-break scenario for its circle. We illustrate in Figure 5.2 an analogous process that to an

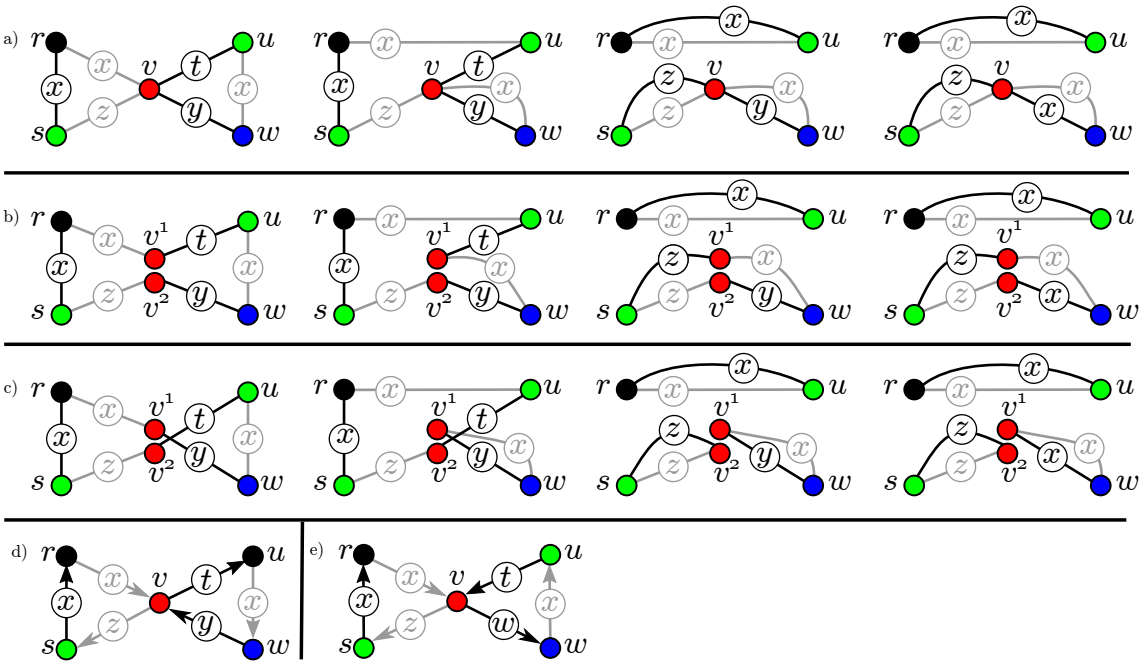


Figure 5.2: In a) an $\mathcal{O}$-scenario ρ for a labeled simple cycle (S, λ) containing a double vertex v is depicted. Vertex colors in the figure represent their labels. ρ is not change-first, as it starts with two $\mathcal{O}$-breaks and finishes with a single $\mathcal{O}$-change. b) depicts a labeled simple cycle $(\hat{S}_1, \hat{\lambda}_1)$ and a sequence $\hat{\rho}_1$ of two $\mathcal{O}$-breaks and an $\mathcal{O}$-change. (S, λ) can be obtained from $(\hat{S}_1, \hat{\lambda}_1)$ by merging the vertices v^1 and v^2 into v. ρ can be obtained from $\hat{\rho}_1$ by replacing the occurrences of v^1 and v^2 with v. The φ-costs of ρ and $\hat{\rho}_1$ are equal as v^1, v^2, and v have equal labels. In d) an Eulerian orientation of (S, λ) is depicted. Its labeled v-circle, the circle obtained from the Eulerian tour starting from v with a black directed edge, is $(\hat{S}_1, \hat{\lambda}_1)$. Transform $(\hat{S}_1, \hat{\lambda}_1)$ by replacing the labeled edges $(\{v^1, u\}, black, t)$ and $(\{v^2, w\}, black, y)$ with $(\{v^2, u\}, black, t)$ and $(\{v^1, w\}, black, y)$ to obtain a simple labeled cycle $(\hat{S}_2, \hat{\lambda}_2)$ presented in d). In the black 2-breaks of $\hat{\rho}_1$ replace the occurrences of v^1 with v^2, and those of v^2 with v^1, to obtain an $\mathcal{O}$-scenario $\hat{\rho}_2$ for $(\hat{S}_2, \hat{\lambda}_2)$ with φ-cost equal to that of ρ. In e) an Eulerian orientation of (S, λ) is depicted whose labeled v-circle is $(\hat{S}_2, \hat{\lambda}_2)$.

$\mathcal{O}$-scenario ρ for (S, λ) assigns a labeled circle $(C, \hat{\lambda})$ of (S, λ) and its $\mathcal{O}$-scenario $\hat{\rho}$ of the same φ-cost as ρ. Choose a labeled Eulerian orientation $(\overrightarrow{C}, \hat{\lambda})$ of $(C, \hat{\lambda})$. For every double vertex v in S, merge the vertices v^1 and v^2 in $(\overrightarrow{C}, \hat{\lambda})$ to obtain a labeled Eulerian orientation $(\overrightarrow{S}, \lambda)$ of (S, λ). From Figure 5.1 it should be clear that the labeled u-circle of $(\overrightarrow{S}, \lambda)$ and $(C, \hat{\lambda})$ are isomorphic. Finally, due to Lemma 29, MCPS-costs of isomorphic labeled graphs are equal. This means that $\varphi(\rho)$ is greater than or equal to MCPS-cost of the labeled u-circle of $(\overrightarrow{S}, \lambda)$. □

5.4 φ-MCPS for a Circle

5.4.1 Introduction

Here we build on Theorem 8 stating that a 2-break scenario for a sub-circle of a circle C can be partitioned into 2-break scenarios for smaller sub-circles of C. We adjust its proof to show that it stays valid for a parsimonious $\mathcal{O}$-scenario on a labeled circle. This partitioning of a parsimonious $\mathcal{O}$-scenario allows for a dynamic programming algorithm for MCPS presented in Subsection 5.4.5.

5.4.2 A Circular Straight Line Embedding of a Circle

Fix a circular straight-line embedding Σ_C of a labeled circle (C, λ) on n vertices. As in Subsection 3.5.3, number the vertices with $[\![1, n]\!]$ while respecting their clockwise order on Σ_C and ensuring that the colored edge going clockwise from n to 1 is gray. For $i, j \in [\![1, n]\!]$ with i and j of different parity, define $C[i, j]$ to be the sub-circle of C consisting of the path going clockwise from i to j in Σ_C and its colored outer edge (see Definition 47) which is $(\{i, j\}, gray)$ if i is odd, and $(\{i, j\}, black)$ otherwise. Denote the color of this edge by $col_{\{i,j\}}$, and the opposite color by $\overline{col_{\{i,j\}}}$.

Definition 72 (x-labeled sub-circle)**.** $(C[i, j], \tilde{\lambda})$ *is a* labeled sub-circle *of* (C, λ)*, if the labels of all the colored edges of* $C[i, j]$ *except* $(\{i, j\}, col_{\{i,j\}})$ *and the labels of its vertices coincide with the labels given by* λ*. If the label of the colored outer edge* $(\{i, j\}, col_{\{i,j\}})$ *of* $C[i, j]$ *is* x*, then* $(C[i, j], \tilde{\lambda})$ *is the* x-labeled sub-circle *of* (C, λ)*, we denote it by* $(C[i, j], \lambda^x)$*.*

5.4.3 Partitioning a Parsimonious $\mathcal{O}$-scenario for a Circle into the Scenarios for its Sub-circles

Definition 73 (outer-change-free scenario for a sub-circle)**.** *An* $\mathcal{O}$*-scenario for a labeled sub-circle* $(C[i, j], \lambda^x)$ *is* outer-change-free *if it does not contain an* $\mathcal{O}$*-change acting on its labeled outer edge* $(\{i, j\}, col_{\{i,j\}}, x)$*.*

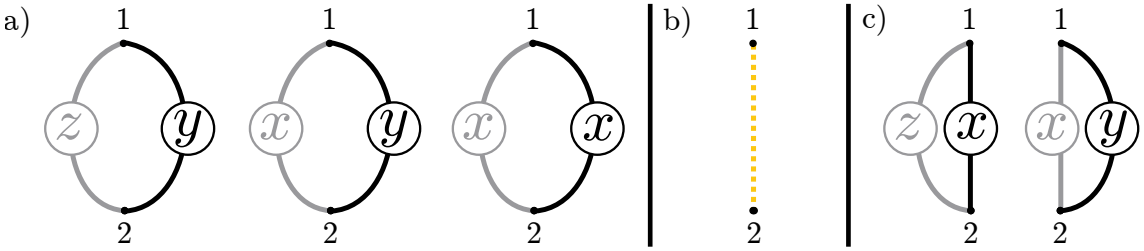


Figure 5.3: A parsimonious change-first $\mathcal{O}$-scenario $\rho = \Big((\{1,2\}, gray, z) \to (\{1,2\}, gray, x), (\{1,2\}, black, y) \to (\{1,2\}, black, x)\Big)$ for a labeled circle (C, λ) having 2 vertices is depicted in a). The underlying 2-break scenario of ρ is empty, as ρ does not contain any $\mathcal{O}$-breaks. The scenario graph and the scenario matching of ρ coincide and are depicted in dashed in b). The labeled subcircles $(C[2,1], \lambda^x)$ and $(C[1,2], \lambda^x)$ of C are depicted in c). ρ can be partitioned into $\rho_1 = \Big((\{1,2\}, gray, z) \to (\{1,2\}, gray, x)\Big)$ and $r_2 = \Big((\{1,2\}, black, y) \to (\{1,2\}, black, x)\Big)$, that are parsimonious change-first outer-change-free scenarios for $(C[2,1], \lambda^x)$ and $(C[1,2], \lambda^x)$ respectively.

Lemma 30. *Take a parsimonious change-first $\mathcal{O}$-scenario ρ for a labeled circle (C, λ) and an edge $\{i, j\}$ of its scenario graph. There exists an edge label x, such that ρ can be partitioned into parsimonious outer-change-free $\mathcal{O}$-scenarios for $(C[i,j], \lambda^x)$ and $(C[j,i], \lambda^x)$.*

Proof. The proof is by induction on the number of vertices in C. If (C, λ) has two vertices, then the scenario graph of ρ contains a single edge and ρ contains at most two $\mathcal{O}$-changes an no $\mathcal{O}$-breaks. The label x of its colored edges obtained after the $\mathcal{O}$-changes are performed satisfies the statement. See Figure 5.3 for an example.

Suppose that the statement is true for any circle with at most $2k-2 \geq 2$ vertices and take a circle C on $2k$ vertices. See Figure 3.9 for an illustration of the proof that follows. Take the first $\mathcal{O}$-break (τ,χ) in ρ. Being change-first, ρ contains at most two $\mathcal{O}$-changes acting on the colored edges replaced by (τ,χ), denote them by μ. Denote by ρ' the subsequence of ρ obtained once μ and (τ,χ) are deleted. Denote by (C', λ') the labeled graph that is obtained from (C,λ) after μ and (τ,χ) are performed. ρ' is a parsimonious $\mathcal{O}$-scenario for (C', λ'). Due to Corollary 2, it has two connected components that are both circles, denote them by (C_1, λ_1) and (C_2, λ_2). Due to Corollary 9, ρ' can be partitioned into ρ_1 and ρ_2, that are parsimonious $\mathcal{O}$-scenarios, change-first by construction, for (C_1, λ_1) and (C_2, λ_2) respectively.

Due to Theorem 7, the scenario graph of ρ inherits from Σ_C a circular straight-line embedding in which all the bounded faces are quadrilaterals. The edges of τ and the edge $\{i, j\}$ all belong to the scenario graph of ρ. This means that both i and j either belong to C_1 or C_2, without loss of generality we can suppose that it is C_2. This also means that all the vertices of τ belong either to $C[i,j]$ or $C[j,i]$.

Without loss of generality we can suppose that it is $C[j,i]$.

The inductive hypothesis holds for a triplet $\Big((C_2,\lambda_2),\rho_2,\{i,j\}\Big)$, providing us with a label x, and a partition of ρ_2 into two parsimonious outer-change-free scenarios for the x-labeled sub-circles of (C_2,λ_2). By construction, one of these labeled sub-circles is $(C[i,j],\lambda^x)$. Denote the other one by (C_5,λ_5) and denote the parsimonious outer-change-free $\mathcal{O}$-scenarios obtained for them by $\rho_{[i,j]}$ and ρ_5 respectively.

By now we know that ρ is a shuffle of μ, (τ,χ), ρ_1, $\rho_{[i,j]}$, and ρ_5. Remove $\rho_{[i,j]}$, μ and (τ,χ) from ρ to obtain a shuffle of ρ_1 and ρ_5, that we denote by $\overline{\rho}$. Due to Observation 21, $\overline{\rho}$ is a parsimonious $\mathcal{O}$-scenario for the union of the vertex-disjoint circles (C_1,λ_1) and (C_5,λ_5), which is also the labeled graph obtained from $(C[j,i],\lambda^x)$ after μ and (τ,χ) are performed. Finally, by deleting only $\rho_{[i,j]}$ from ρ, we obtain a shuffle of μ, (τ,χ) and $\overline{\rho}$, that is a parsimonious $\mathcal{O}$-scenario for $(C[j,i],\lambda^x)$, that does not contain any $\mathcal{O}$-changes acting on its colored outer edge, and together with $\rho_{[i,j]}$ satisfies the statement. □

Definition 74 (Matched scenario for a sub-circle)**.** *A parsimonious $\mathcal{O}$-scenario for a sub-circle $C[i,j]$ is* matched*, if its scenario matching includes an edge $\{i,j\}$. It is* non-matched *otherwise.*

Theorem 15. *Take ρ, a parsimonious matched outer-change-free change-first $\mathcal{O}$-scenario for $(C[i,j],\lambda^x)$ with $i+3\leq j$. ρ can be partitioned into an $\mathcal{O}$-break $(\tau,\chi)=\Big(\Big\{(\{i,k\},z),(\{l,j\},t)\Big\}\rightarrow\Big\{(\{i,j\},x),(\{k,l\},y)\Big\},\overline{col_{\{i,j\}}}\Big)$ with $i<k<l<j$, and parsimonious outer-change-free change-first $\mathcal{O}$-scenarios for $(C[i,k],\lambda^z)$, $(C[k,l],\lambda^y)$, and $(C[l,j],\lambda^t)$ with the $\mathcal{O}$-scenarios for $(C[i,k],\lambda^z)$ and $(C[l,j],\lambda^t)$ being matched.*

Proof. Denote the underlying 2-break scenario of ρ by ρ^u. See Figure 5.4 for an example of ρ^u. Due to Theorem 8, ρ^u contains a 2-break $\tau=\Big(\Big\{\{i,k\},\{l,j\}\Big\}\rightarrow\Big\{\{i,j\},\{k,l\}\Big\},\overline{col_{\{i,j\}}}\Big)$ with $i<k<l<j$. This means that ρ contains an $\mathcal{O}$-break (τ,χ) with a labeling χ of τ edges. ρ is change-first, thus it does not contain an $\mathcal{O}$-change modifying the labeled edge $(\{i,j\},\overline{col_{\{i,j\}}},\chi(\{i,j\}))$ introduced by (τ,χ). ρ is outer-change-free, thus it also does not contain an $\mathcal{O}$-change modifying its labeled outer edge $(\{i,j\},col_{\{i,j\}},x)$. Due to Theorem 7, τ is a single 2-break in ρ^u with $\{i,j\}$ among its edges. This means that ρ does not contain an $\mathcal{O}$-break replacing labeled edge $(\{i,j\},\overline{col_{\{i,j\}}},\chi(\{i,j\}))$ or $(\{i,j\},col_{\{i,j\}},x)$. Thus ρ transforms $(C[i,j],\lambda^x)$ into a terminal graph containing labeled edges $(\{i,j\},\overline{col_{\{i,j\}}},\chi(\{i,j\}))$ and $(\{i,j\},col_{\{i,j\}},x)$, which means that $\chi(\{i,j\})=x$.

Apply Lemma 30 for $\{i,k\},\{k,l\}$ and $\{l,j\}$ to obtain edge labels z,t,y and parsimonious outer-change-free $\mathcal{O}$-scenarios ρ_1,ρ_2, and ρ_3 for sub-circles $(C[i,k],\lambda^z)$, $(C[k,l],\lambda^y)$ and $(C[l,j],\lambda^t)$, that together with (τ,χ) partition ρ. $\chi(\{i,k\})=z$, $\chi(\{k,l\})=y$ and $\chi(\{l,j\})=t$, since ρ is change-first. All we have to prove now is that ρ_1 and ρ_3 are matched.

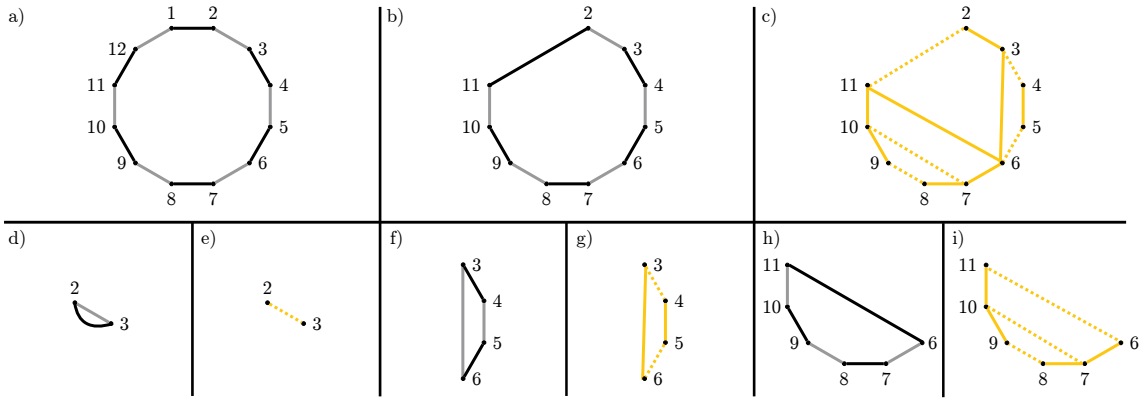


Figure 5.4: A circular straight-line embedding Σ_C of a circle C is depicted in a), while b) presents its sub-circle $C[2, 11]$. In c) the scenario graph and the matching of an underlying parsimonious 2-break scenario ρ^u for $C[2, 11]$ is depicted, with $\rho = \Big(\Big(\big\{\{7, 8\}, \{9, 10\}\big\} \to \big\{\{7, 10\}, \{8, 9\}\big\}, black\Big), \Big(\big\{\{6, 7\}, \{10, 11\}\big\} \to \big\{\{6, 11\}, \{7, 10\}\big\}, gray\Big), \Big(\big\{\{2, 3\}, \{6, 11\}\big\} \to \big\{\{2, 11\}, \{3, 6\}\big\}, gray\Big), \Big(\big\{\{3, 6\}, \{4, 5\}\big\} \to \big\{\{3, 4\}, \{5, 6\}\big\}, gray\Big)\Big)$. ρ^u is matched and contains a 2-break $\tau = \Big(\big\{\{2, 3\}, \{6, 11\}\big\} \to \big\{\{2, 11\}, \{3, 6\}\big\}, gray\Big)$ introducing a colored edge $(\{2, 11\}, gray)$. ρ^u can be partitioned into τ and parsimonious scenarios ρ^u_1, ρ^u_2 and ρ^u_3 for sub-circles $C[2, 3], C[3, 6]$, and $C[6, 11]$. These sub-circles and the corresponding scenario graphs are depicted in d)-i). ρ^u_1 is empty, thus matching by definition. $\rho^u_2 = \Big(\Big(\big\{\{3, 6\}, \{4, 5\}\big\} \to \big\{\{3, 4\}, \{5, 6\}\big\}, gray\Big)\Big)$ is a non-matching 2-break scenario for $C[3, 6]$, while $\rho^u_3 = \Big(\Big(\big\{\{7, 8\}, \{9, 10\}\big\} \to \big\{\{7, 10\}, \{8, 9\}\big\}, black\Big), \Big(\big\{\{6, 7\}, \{10, 11\}\big\} \to \big\{\{6, 11\}, \{7, 10\}\big\}, gray\Big)\Big)$ is a matching 2-break scenario for $C[5, 11]$.

If $k = i + 1$, then ρ_1 is an $\mathcal{O}$-scenario for a labeled circle on two vertices and is matched by definition. Suppose that $i \leq k + 3$, meaning that a colored edge $(\{i, k\}, \overline{col_{\{i,j\}}})$ is not present in C. ρ^u contains a 2-break replacing it, thus due to Observation 1, it must also contain a 2-break introducing it. ρ^u is a parsimonious 2-break scenario, thus due to Theorem 7, it contains only two 2-breaks with $\{i, k\}$ among their edges. Due to the same theorem, the underlying 2-break scenario of ρ_1 contains a single 2-break with $\{i, k\}$ among its edges. That 2-break is not τ, which means that it introduces $(\{i, k\}, \overline{col_{\{i,j\}}})$, and thus ρ_1 is indeed a matched $\mathcal{O}$-scenario. The same analysis applies to ρ_3. □

The proof of the following theorem for non-matched $\mathcal{O}$-scenarios is analogous to that of Theorem 15.

Theorem 16. *Take a parsimonious non-matched outer-change-free change-first $\mathcal{O}$-scenario ρ for $(C[i, j], \lambda^x)$ with $i + 3 \leq j$. There exists an $\mathcal{O}$-break $(\tau, \chi) =$*

$\left(\left\{(\{i,j\},x),(\{k,l\},y)\right\} \rightarrow \left\{(\{i,k\},z),(\{l,j\},t)\right\}, col_{\{i,j\}}\right)$ *with* $i < k < l < j$, *such that* ρ *can be partitioned into* (τ, χ) *and parsimonious outer-change-free change-first* $\mathcal{O}$*-scenarios for* $(C[i,k], \lambda^z)$, $(C[k,l], \lambda^y)$, *and* $(C[l,j], \lambda^t)$ *with the* $\mathcal{O}$*-scenario for* $(C[k,l], \lambda^y)$ *being matched.*

5.4.4 Minimal Weight Polygon Quadrangulation

Our algorithm to be presented in Subsection 5.4.5 can be seen as a generalization of that for the Minimal Weight Polygon Quadrangulation problem (MWPQ) presented by Massarwi, Sosin, and Elber in Section 3.1.2 of [72]. There arbitrary positive weights are assigned to the quadrilaterals in a polygon, and the authors ask for a quadrangulation minimizing the sum of the weights of its quadrilaterals.

A quadrangulation Q of a sub-circle $C[i,j]$ contains a quadrilateral f incident to the edge $\{i,j\}$. Suppose that its vertices are $i < k < l < j$. See Figure 5.5 for an example. The weight of Q is equal to the sum of the weights of Q restricted to $C[i,k]$, Q restricted to $C[k,l]$, Q restricted to $C[l,j]$ and the weight of f. This partition allows for an $O(n^4)$ time dynamic programming algorithm for MWPQ, which is very similar to an $O(n^3)$ time dynamic programming algorithm for the Minimal Weight Triangulation problem on a simple polygon presented by Klincsek [67].

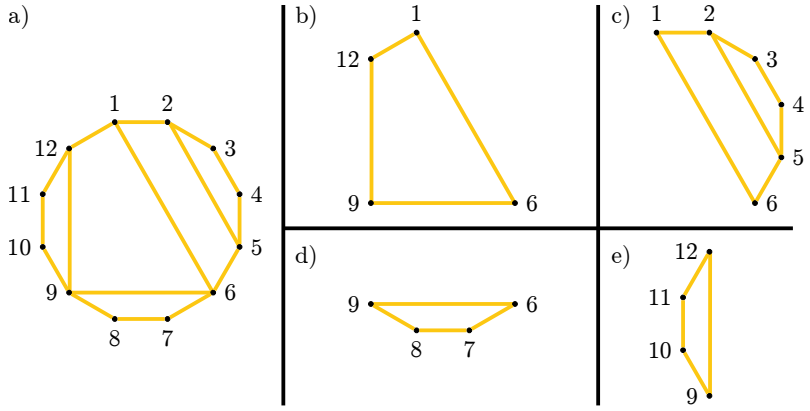


Figure 5.5: A quadrangulation Q of $C[1,12]$ is depicted in a). It contains a quadrilater f incident to edge $\{1,12\}$ that is depicted in b). When restricted to the sub-circles $C[1,6]$, $C[6,9]$ and $C[9,12]$, Q provides quadrangulations depicted in c), d) and e).

5.4.5 A Dynamic Programming Algorithm for φ-MCPS for a Circle

Introduction and Notations

We use Theorem 15 and Theorem 16 to express the minimum φ-cost of a parsimonious outer-change-free matched (or non-matched) $\mathcal{O}$-scenario for a labeled sub-circle as a sum of these costs for smaller labeled sub-circles. Once these costs are obtained for $(C[1,n],\lambda^x)$ for every edge label x, we combine them to obtain the minimum φ-cost of a parsimonious $\mathcal{O}$-scenario for (C,λ).

Take a complete $(\mathcal{O},\varphi)$ and an embedding Σ_C of a labeled circle (C,λ) on n vertices as in Subsection 5.4.2. Denote by $\varphi_m[i,j,x]$ (respectively $\varphi_u[i,j,x]$) the minimum φ-cost of a parsimonious outer-change-free matched (respectively non-matched) $\mathcal{O}$-scenario if it exists, otherwise define $\varphi_m[i,j,x]=\infty$ (respectively $\varphi_u[i,j,x]=\infty$). Denote by $\varphi[i,j,x]$ the minimum φ-cost of a parsimonious outer-change-free $\mathcal{O}$-scenario if it exists, otherwise define $\varphi[i,j,x]=\infty$. By construction $\varphi[i,j,x]=\min(\varphi_m[i,j,x],\varphi_u[i,j,x])$. Take a labeled edge $(\{i,i+1\},col,x)$ present in (C,λ) and an edge label y with a and b being the labels of i and $i+1$. Define $W^{col}_{\{a,b\}}(x,x)=0$. If an edge-label change $\Big((\{a,b\},x);(\{a,b\},y);col\Big)$ is in $\mathcal{O}$, then denote $\varphi\Big((\{a,b\},x);(\{a,b\},y);col\Big)$ by $W^{col}_{\{a,b\}}(x,y)$, otherwise, define $W^{col}_{\{a,b\}}(x,y)=\infty$. Do the same for a labeled edge $(\{n,1\},gray,x)$ in (C,λ).

Initializing φ_m and φ_u

We start by initializing φ_u and φ_m for the labeled sub-circles of (C,λ) on two vertices.

Lemma 31. *Take a labeled edge $(\{i,i+1\},col,x)$ of (C,λ) and an edge label y. $\varphi_u[i,i+1,y]=\infty$ and $\varphi_m[i,i+1,y]=W^{col}_{\{a,b\}}(x,y)$, with a and b being the labels of i and $i+1$.*

Proof. $C[i,i+1]$ has two vertices. The matching of a parsimonious $\mathcal{O}$-scenario for $(C[i,i+1],\lambda^y)$ contains edge $\{i,i+1\}$ by construction, thus $\varphi_u[i,i+1,y]=\infty$. The labeled edges of $(C[i,i+1],\lambda^y)$ are $(\{i,i+1\},col,x)$ and $(\{i,i+1\},\overline{col},y)$. If a parsimonious outer-change-free $\mathcal{O}$-scenario for $(C[i,i+1],\lambda^y)$ exists, then it is a sequence of $\mathcal{O}$-changes transforming a labeled edge $(\{i,i+1\},col,x)$ into $(\{i,i+1\},col,y)$. $(\mathcal{O},\varphi)$ is complete, thus if it does not exist, then $\varphi_m[i,i+1,y]=W^{col}_{\{a,b\}}(x,y)=\infty$. Suppose that such an $\mathcal{O}$-scenario exists and take a parsimonious outer-change-free $\mathcal{O}$-scenario for $(C[i,i+1],\lambda^y)$ with φ-cost equal to $\varphi_m[i,i+1,y]$. Due to Lemma 25 there exists a parsimonious change-first outer-change-free $\mathcal{O}$-scenario of the same φ-cost. It consists of a single $\mathcal{O}$-change, thus $\varphi_m[i,i+1,y]=W^{col}_{\{a,b\}}(x,y)$. □

Corollary 10. *Take a labeled edge $(\{n,1\},gray,x)$ of (C,λ) and an edge label y.*

$\varphi_u[n,1,y] = \infty$ *and* $\varphi_m[n,1,y] = W^{gray}_{\{a,b\}}(x,y)$*, with* a *and* b *being the labels of* n *and* 1*.*

Computing φ_m and φ_u

Now we show how to compute φ_m and φ_u for a labeled sub-circle of (C,λ) using φ_m and φ_u values for its smaller labeled sub-circles. We have demonstrated in Theorem 15 that a matched outer-change-free change-first $\mathcal{O}$-scenario for $(C[i,j],\lambda^x)$ contains an $\mathcal{O}$-break introducing a colored edge $(\{i,j\},x,\overline{col_{\{i,j\}}})$. This $\mathcal{O}$-break has the following form.

Definition 75 (matched $\mathcal{O}$-break for a labeled sub-circle)**.** *An* $\mathcal{O}$*-break* (τ,χ) *is* matched for $(C[i,j],\lambda^x)$*, if* $\tau = \Big(\Big\{\{i,k\},\{l,j\}\Big\} \to \Big\{\{i,j\},\{k,l\}\Big\},\overline{col_{\{i,j\}}}\Big)$ *for* $i < k < l < j$ *with* $i \equiv l \mod 2$ *and* $k \equiv j \mod 2$*, and* $\chi(\{i,j\}) = x$*.*

In the following lemma we iterate through the space of the matched $\mathcal{O}$-breaks for $(C[i,j],\lambda^x)$ to compute $\varphi_m[i,j,x]$.

Lemma 32. *For* $i,j \in [\![1,n]\!]$ *with* $j \geq i+3$ *of different parity and an edge label* x *we have:*

$$\varphi_m[i,j,x] = \min_{(\tau,\chi)\in M}\Big(\varphi_m[i,k,z] + \varphi_m[l,j,t] + \varphi[k,l,y] + \varphi(\tau,\chi)\Big),$$

where M *is a set of matched* $\mathcal{O}$*-breaks for* $(C[i,j],\lambda^x)$*.*

Proof. First we show that $\varphi_m[i,j,x]$ is greater than or equal to the proposed expression. If $\varphi_m[i,j,x] = \infty$, then this inequality is trivially true. Otherwise, due to Lemma 25, there exists a parsimonious change-first outer-change-free matched $\mathcal{O}$-scenario ρ for $(C[i,j],\lambda^x)$ with φ-cost equal to $\varphi_m[i,j,x]$. Due to Theorem 15, it can be partitioned into a matched $\mathcal{O}$-break $(\tau,\chi) = \Big(\Big\{(\{i,k\},z),(\{l,j\},t)\Big\} \to \Big\{(\{i,j\},x),(\{k,l\},y)\Big\},\overline{col_{\{i,j\}}}\Big)$ and parsimonious outer-change-free $\mathcal{O}$-scenarios ρ_1, ρ_2 and ρ_3 for the sub-circles $(C[i,k],\lambda^z)$, $(C[k,l],\lambda^y)$ and $(C[l,j],\lambda^t)$, with ρ_1 and ρ_3 being matched. This establishes an inequality $\varphi_m[i,j,x] \geq \varphi(\tau,\chi) + \varphi_m[i,k,z] + \varphi_m[l,j,t] + \varphi[k,l,y]$.

Now we show that the opposite inequality holds. Take a matched $\mathcal{O}$-break $(\tau,\chi) = \Big(\Big\{(\{i,k\},z),(\{l,j\},t)\Big\} \to \Big\{(\{i,j\},x),(\{k,l\},y)\Big\},\overline{col_{\{i,j\}}}\Big)$. If $\varphi_m[i,k,z] + \varphi_m[l,j,t] + \varphi[k,l,y] + \varphi(\tau,\chi) = \infty$, then this inequality is trivially true. Otherwise take parsimonious outer-change-free $\mathcal{O}$-scenarios ρ_1, ρ_2, and ρ_3 of φ-costs $\varphi_m[i,k,z]$, $\varphi[k,l,y]$, and $\varphi_m[l,j,t]$, for the labeled sub-circles $(C[i,k],\lambda^z)$, $(C[k,l],\lambda^y)$, and $(C[l,j],\lambda^t)$ respectively, with ρ_1 and ρ_3 being matched $\mathcal{O}$-scenarios. The sequence obtained by performing ρ_1 and ρ_3, followed by (τ,χ) and ρ_2 is a parsimonious matched outer-change-free $\mathcal{O}$-scenario for $(C[i,j],\lambda^x)$. Thus establishing $\varphi_m[i,j,x] \leq \varphi(\tau,\chi) + \varphi_m[i,k,z] + \varphi_m[l,j,t] + \varphi[k,l,y]$. □

Analogously, we introduce non-matched $\mathcal{O}$-breaks and use Theorem 16 to establish Lemma 33.

Definition 76 (non-matched $\mathcal{O}$-break for a labeled sub-circle)**.** *An $\mathcal{O}$-break (τ, χ) is* matched for $(C[i,j], \lambda^x)$*, if $\Big(\big\{\{i,j\},\{k,l\}\big\} \to \big\{\{i,k\},\{l,j\}\big\}, col_{\{i,j\}}\Big)$ for $i < k < l < j$ with $i \equiv l \mod 2$ and $k \equiv j \mod 2$, and $\chi(\{i,j\}) = x$.*

Lemma 33. *For $i, j \in [\![1,n]\!]$ with $j \geq i+3$ of different parity and an edge label x we have:*

$$\varphi_u[i,j,x] = \min_{(\tau,\chi)\in M} \Big(\varphi[i,k,z] + \varphi[l,j,t] + \varphi_m[k,l,y] + \varphi(\tau,\chi)\Big),$$

where M is a set of non-matched $\mathcal{O}$-breaks for $(C[i,j], \lambda^x)$.

Computing MCPS$_\varphi$(C,λ)

Finally we use values $\varphi_m[1,n,y]$ and $\varphi_u[1,n,y]$ to compute $\mathrm{MCPS}_\varphi(C,\lambda)$.

Lemma 34. *Denote the labels of the vertices 1 and n, by a and b, and denote the label of the colored edge $(\{1,n\}, gray)$ by x. $MCPS_\varphi(C,\lambda)$ for a labeled circle is equal to:*

$$\min\Big(\min_{y\in\Sigma_E}\Big(\varphi_m[1,n,y] + W^{gray}_{\{a,b\}}(x,y)\Big), \min_{y\in\Sigma_E}\Big(\varphi_u[1,n,y] + W^{gray}_{\{a,b\}}(x,y)\Big)\Big)$$

Lemma 35. *Denote the labels of the vertices 1 and n, by a and b, and denote the label of the colored edge $(\{1,n\}, gray)$ by x. $MCPS_\varphi(C,\lambda)$ for a labeled circle is equal to $\min_{y\in\Sigma_E}\Big(\varphi[1,n,y] + W^{gray}_{\{a,b\}}(x,y)\Big)$*

Proof. We first show that $\mathrm{MCPS}_\varphi(C,\lambda)$ is larger than or equal to the proposed expression. If $\mathrm{MCPS}_\varphi(C,\lambda) = \infty$, then this inequality is trivially true. Otherwise due to Corollary 8 and the completeness of $(\mathcal{O},\varphi)$ there exists a change-first φ-MCPS $\mathcal{O}$-scenario ρ for (C,λ). Such a ρ contains at most one $\mathcal{O}$-change that modifies a labeled edge $(\{n,1\}, gray, x)$. If such an $\mathcal{O}$-change does not exist, then ρ is outer-change-free and $\mathrm{MCPS}_\varphi(C,\lambda)$ is larger than or equal to $\varphi[1,n,x]$. If such an $\mathcal{O}$-change exists, then denote it by μ and denote the label it provides by x'. ρ with μ removed is a parsimonious outer-change-free $\mathcal{O}$-scenario for $(C[1,n], \lambda^{x'})$. This means that $\mathrm{MCPS}_\varphi(C,\lambda) \geq \varphi[1,n,x'] + W^{gray}_{\{a,b\}}(x,x') \geq \min_{y\in\Sigma_E}\Big(\varphi[1,n,y] + W^{gray}_{\{a,b\}}(x,y)\Big)$

If the proposed expression is equal to ∞, then there is nothing more to prove. Otherwise, take an edge label x' together with a parsimonious outer-change-free $\mathcal{O}$-scenario ρ for $(C[1,n], \lambda^{x'})$ realizing the minimum of the expression. Add an $\mathcal{O}$-change $(\{1,n\}, gray, x) \to (\{1,n\}, gray, x')$ to the beginning of ρ to obtain a parsimonious $\mathcal{O}$-scenario for (C,λ). This way we obtain $\mathrm{MCPS}_\varphi(C,\lambda) \leq \varphi[1,n,x'] + W^{gray}_{\{a,b\}}(x,x') = \min_{y\in\Sigma_E}\Big(\varphi[1,n,y] + W^{gray}_{\{a,b\}}(x,y)\Big)$. □

The MCPS Problem

Theorem 17. *The MCPS problem for a complete $(\mathcal{O}, \varphi)$ with L edge labels and a labeled circle (C, λ) on n vertices can be solved in $O(n^4L^4)$ worst case time.*

Proof. φ_m and φ_u can be initialized for n colored edges of C and L edge labels in $O(nL)$ time. Then $\varphi_m[i, j, x]$ and $\varphi_u[i, j, x]$ for $O(n^2L)$ combinations of vertices and edge labels are computed. For a fixed triplet $[i, j, x]$ one has to iterate through at most $O(n^2L^3)$ matched and non-matched $\mathcal{O}$-breaks. If an individual $\mathcal{O}$-break can be checked in constant time, then this results in $O(n^4L^4)$ time complexity. The last step of computing $\text{MCPS}_\varphi(C, \lambda)$ can be performed in $O(L)$ time. □

Corollary 11. *The algorithm can be easily parallelized. Take $d \in [\![1, n-1]\!]$. The values $\varphi_m[i, i+d, x]$ and $\varphi_u[i, i+d, x]$ for every $i \in [\![1, n-d]\!]$ and every edge label x can be computed in parallel.*

5.5 φ-MCPS for a Breakpoint Graph

In this section we show that the φ-MCPS problem for a labeled breakpoint graph can be solved using an algorithm for φ-MCPS on a labeled circle as a subroutine. The latter might be either the general dynamic programming algorithm from the previous section, or an algorithm with a lower time complexity specifically tailored for a particular cost function as in [95, 27].

A breakpoint graph, as defined in Section 3.3, is a graph with at most one vertex of black and gray degrees larger than 1. Denote this vertex by $\circ$. An AA path (respectively a BB path) of a breakpoint graph is an alternating tour of odd length starting from $\circ$ with a black edge (respectively gray edge). A simple subgraph of a breakpoint graph is either a circle or a union of an AA and a BB path. Denote by $\mathcal{B}(G)$ a complete bipartite graph having the AA and the BB paths of G as vertices. Lemma 14 states that a Maximum Alternating Edge-disjoint Cycle Decomposition of a breakpoint graph G can be identified with a perfect matching of $\mathcal{B}(G)$ plus the circle subgraphs of G.

Take a pair $(\mathcal{O}, \varphi)$ and a labeled breakpoint graph (G, λ) on n vertices. Upon inspection of Figure 5.6 it should be clear that a labeled simple subgraph (S, λ_S) of (G, λ) has at most four labeled Eulerian orientations and two non-isomorphic labeled circles. Due to Theorem 14, $\text{MCPS}_\varphi(S, \lambda_S)$ is equal to the minimum of the MCPS_φ-costs of these labeled circles. Use an algorithm for φ-MCPS on a labeled circle to compute the MCPS_φ-costs of all the simple labeled subgraphs of (G, λ). Weight the edges of $\mathcal{B}(G)$ with the MCPS_φ-costs of the corresponding simple labeled subgraphs. Due to Lemma 14 and Theorem 13, $\text{MCPS}_\varphi(G, \lambda)$ is equal to the sum of the MCPS_φ-costs of the labeled circle subgraphs of (G, λ) plus the minimum weight of a perfect matching of $\mathcal{B}(G)$.

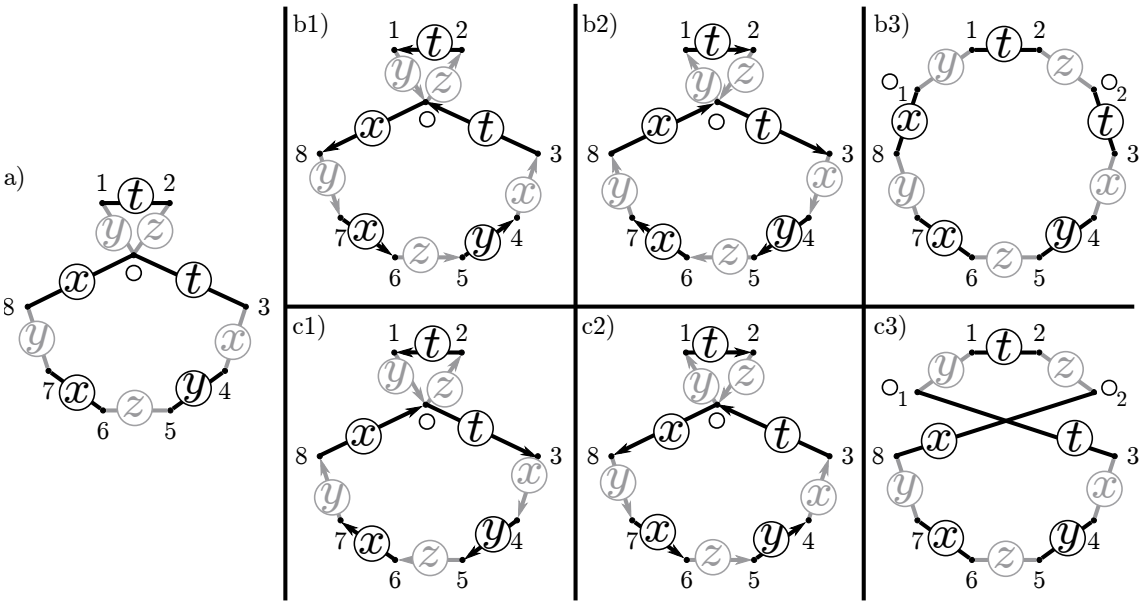


Figure 5.6: A labeled simple cycle of a labeled breakpoint graph is depicted in a). Its four labeled Eulerian orientations are given in b1), b2), c1) and c2). By inverting the directions of the labeled edges of the labeled Eulerian orientation in b1) we obtain that in b2). Their labeled 1-circles are isomorphic, and that of b1) is depicted in b3). The same is true for the figures in c1), c2) and c3). Due to reasons explained in Theorem 14, every labeled circle of this labeled simple cycle is isomorphic to either a labeled circle in b3) or that in c3).

G has $O(n)$ AA and BB paths forming $O(n^2)$ simple non-circle subgraphs each containing $O(n)$ edges. If the MCPS$_\varphi$-cost of a labeled circle with r edges can be computed in $O(r^t)$ time for some constant $t \geq 1$, then $\mathcal{B}(G)$ edges can be weighted in $O(n^{t+2})$ time. The minimum weight perfect matching of $\mathcal{B}(G)$ can be found in $O(n^3)$ time using the Hungarian algorithm, which leads to an $O(n^{t+2})$ time algorithm for the φ-MCPS problem on a labeled breakpoint graph. In what follows we use amortized analysis to show that its time complexity is actually $O(n^{t+1} + n^3)$.

Denote by p and c the number of AA paths and circle subgraphs in a breakpoint graph G. By construction, G has an equal number of AA and BB paths. Denote by P the number of colored edges of G that belong to an AA or a BB path. Denote by Q the number of its colored edges that belong to a circle subgraph.

Theorem 18. *For some function f and an $O(f(r))$ time algorithm for φ-MCPS on a labeled circle with r colored edges, there exists an $O(p^2 f(P) + p^3 + cf(Q))$ time algorithm for φ-MCPS on a labeled breakpoint graph. If $f(r) = O(r^t)$ for some constant $t \geq 1$, then the MCPS$_\varphi$-cost of a labeled breakpoint graph can be computed in $O(pP^t + p^3 + Q^t)$ time, which is $O(n^{t+1} + n^3)$.*

Proof. Take a labeled breakpoint graph (G, λ). It has $P + Q$ colored edges parti-

tioned between AA paths, BB paths and circle subgraphs of G that can be easily identified in $O(P+Q)$ time. Due to Theorem 14, $\text{MCPS}_\varphi(S, \lambda_S)$ is equal to the minimum of the MCPS_φ-costs of at most two non-isomorphic labeled circles containing $O(P)$ edges. This means that the p^2 edges of $\mathcal{B}(G)$ can be weighted in $O(p^2 f(P))$ time. A minimum weight perfect matching of $\mathcal{B}(G)$ can be found in $O(p^3)$ time using the Hungarian algorithm as it has $2p$ vertices. The MCPS_φ-costs of the circle subgraphs of G can be computed in $O(cf(Q))$ time. Combining these results we obtain an $O(p^2 f(P) + p^3 + cf(Q))$ time algorithm for φ-MCPS on (G, λ).

Now suppose that $f(r) = O(r^t)$ for some constant $t \geq 1$. This means that φ-MCPS for a labeled circle with r edges can be computed in $\frac{k}{2}r^t$ steps for some constant k. Let $(d_1, \ldots, d_c)$ denote the numbers of colored edges in the circle subgraphs of G. By definition $\sum_{i=0}^{c} d_i = Q$. The MCPS_φ-cost of these c circles can be computed in $\sum_{i=0}^{c} \frac{k}{2} d_i^t \leq \frac{2}{4} Q^t$ steps. Let $(a_1, \ldots, a_p)$ and $(b_1, \ldots, b_p)$ denote the numbers of the colored edges in the AA and the BB paths. Denote $\sum_{i=0}^{p} a_i$ by P_A and $\sum_{j=0}^{p} b_j$ by P_B. By definition $P = P_A + P_B$. As explained above, the MCPS_φ-cost of a union of an AA path and a BB path having a and b colored edges can be computed in at most $k(a+b)^t$ steps by computing the MCPS_φ-costs of at most two labeled circles with $a + b$ colored edges. The MCPS_φ-cost of every pair of an AA and a BB path can be computed in a number of steps bounded by:

$$\sum_{i=0}^{p}\sum_{j=0}^{p} k(a_i + b_j)^t = k\sum_{i=0}^{p}\sum_{j=0}^{p}\sum_{l=0}^{t}\binom{t}{l} a_i^l b_j^{t-l} = k\sum_{l=0}^{t}\binom{t}{l}\sum_{i=0}^{p}\sum_{j=0}^{p} a_i^l b_j^{t-l}$$

$$= k\sum_{j=0}^{p}\sum_{i=0}^{p} b_j^t + k\sum_{i=0}^{p}\sum_{j=0}^{p} a_i^t + k\sum_{l=1}^{t-1}\binom{t}{l}\sum_{i=0}^{p} a_i^l \sum_{j=0}^{p} b_j^{t-l}$$

$$= kp\sum_{j=0}^{p} b_j^t + kp\sum_{i=0}^{p} a_i^t + k\sum_{l=1}^{t-1}\binom{t}{l}\sum_{i=0}^{p} a_i^l \sum_{j=0}^{p} b_j^{t-l}$$

$$\leq kp(\sum_{j=0}^{p} b_j)^t + kp(\sum_{i=0}^{p} a_i)^t + k\sum_{l=1}^{t-1}\binom{t}{l}(\sum_{i=0}^{p} a_i)^l (\sum_{j=0}^{p} b_j)^{t-l}$$

$$\leq k(pP_B^t + pP_A^t) + pk\sum_{l=1}^{t-1}\binom{t}{l} P_B^{t-l} P_A^l = kp(P_B + P_A)^t = kpP^t$$

Thus $\mathcal{B}(G)$ edges can be weighted in $O(pP^t)$ time. This provides us with an $O(pP^t + p^3 + Q^t)$ time algorithm for computing the MCPS_φ-cost of a labeled breakpoint graph. p, P and Q are $O(n)$, thus the worst case time complexity is $O(n^{t+1} + n^3)$. □

5.6 Time Complexity

We have treated the MCPS problem by proposing a polynomial time dynamic programming algorithm solving it for a labeled circle and an ILP solving it for any

labeled graph. The $O(n^4L^4)$ time complexity of our algorithm for a labeled circle, where n is the number of vertices and L is the number of edge labels, might seem prohibitively high. However this worst case time complexity can be avoided in practice, at least when dealing with the biological data related to genome rearrangements.

To begin with, n remains quite low in the genome breakpoint graphs of interest. In our study [89] n did not exceed 250 even for the species as distant as human and chicken, while for human and gibbon, a pair of species of a particular interest for the study of genome rearrangements [68], n was found to be 80 or less. An initial version of our algorithm, where $L = 1$ and only black 2-breaks are allowed, took seconds to finish for the genome breakpoint graph of human and gibbon and minutes for that of human and chicken. This indicates that for low L values our algorithm stays feasible for any pair $(\mathcal{O}, \varphi)$.

An $\mathcal{O}$-break includes 4 vertices and 4 edge labels, thus the total number B of the $\mathcal{O}$-breaks that might appear in a parsimonious $\mathcal{O}$-scenario for a labeled circle is $O(n^4L^4)$. In our algorithm we check every $\mathcal{O}$-break exactly once, thus its time complexity can be brought down to $O(n^4 + B)$. This can be done with the help of an appropriately chosen data structure for $\mathcal{O}$, where for every quadrilateral we can query the $\mathcal{O}$-breaks with the edges bounding that quadrilateral. Take for example the set of valid operations $\mathcal{O}$ used in Minimum Local Parsimonious Scenario [95] and introduced in Subsection 4.3.2. Here $O(B)$ is actually $O(n^6)$, as every $\mathcal{O}$-break in a parsimonious $\mathcal{O}$-scenario includes only two edge labels that must be already present among the n edge labels of (C, λ).

Finally, once a particular pair $(\mathcal{O}, \varphi)$ is chosen, our algorithm can be used as a blueprint to devise a more efficient algorithm specifically tailored for this pair. For example the φ-costs of some $\mathcal{O}$-breaks might be too high for them to appear in a φ-MCPS $\mathcal{O}$-scenario, or the vertex and edge labels of a sub-circle $C[i, j]$ might impose constraints on the possible labels of its outer colored edge, so we do not have to check L different options.

Chapter 6

Conclusion

We have established novel links between various permutation, string, genome and graph sorting problems, that ultimately led to our framework for cost constrained 2-break scenarios. Even if it is beyond the scope of this study to fully explore the implications of these links, we have briefly mentioned some open optimization problems that might benefit from our observations. These include Swap Median Permutation, DCJ Closest Genome, Token Swapping on trees, and a conjecture stating that there always exists an $O(n^2)$ length transposition decomposition of minimum cost [42]. We have not even touched upon questions related to counting, sampling and estimating, that also hold a number of important open problems. For example, the problem of counting the parsimonious DCJ scenarios transforming one single copy genome into another is conjectured to be #P-complete [75]. Finally, we believe that further investigation might reveal important connections between the study of genome rearrangements, rank aggregation [6], and permutation codes [104] that are not yet fully appreciated.

We have furthered our understanding of parsimonious DCJ scenarios by revealing novel links between parsimonious 2-break scenarios, Minimum Length Transposition Decompositions and quadrangulations of regular polygons. We have defined two scenarios to be equivalent if they consist of the same rearrangements, and described the equivalence classes for this relation. There are $(n+1)^{(n-1)}$ [81] parsimonious black-2-break scenarios for a circle on $2n+2$ vertices, while the number of their equivalence classes is only $\frac{1}{2n+1}\binom{3n}{n} \approx \frac{1}{n^{3/2}}(\frac{27}{4})^n$. This drastic reduction in the search space was instrumental in our work on cost constrained parsimonious scenarios. The true evolutionary scenario, however, is likely to be non-parsimonious [20, 2]. This prompts us to relax the parsimony criterion and the first step in this direction could be a description of the equivalence classes of the 2-break scenarios that are almost parsimonious.

We have introduced a framework for cost constrained DCJs, and demonstrated that within it a particular optimization problem can be efficiently solved. Through-

out this work we have searched to minimize the sum of rearrangement costs, however, our algorithms can be easily modified to maximize their product. If the labels of the intergenes were somehow linked with the probabilities for the rearrangements to occur, then our work could be used to find the most likely parsimonious rearrangement scenarios.

We have shown that a Minimum Cost Parsimonious Scenario can be solved in polynomial time for an arbitrary cost function. The next step is to move from arbitrary to *biologically relevant* costs. Defining the latter term would require a separate project, however for now we say that these are the cost functions for which an MCPS scenario *resembles* the true evolutionary one. In [89] we briefly discussed an idea for testing this *relevance* with the help of what we call *sure* 2-breaks. These are 2-breaks that appear in every parsimonious 2-break scenario for a graph. Take a circle subgraph on four vertices in a genome breakpoint graph G. Every parsimonious 2-break scenario for G necessarily contains either the black or the gray 2-break sorting this circle. This means that if the true evolutionary scenario is *close* to being parsimonious, then it is *likely* to contain one of these 2-breaks, that we call *sure*. A biologically relevant cost function then would be expected to have *low* value for at least one of these *sure* 2-breaks, however statistical tests that would accompany this idea are yet to be developed. In [89] we have shown that a parsimonious 2-break scenario for human and gibbon has ~30 sure 2-breaks that comprise ~25% of its length. By comparing multiple pairs of species we could accumulate a larger set of sure 2-breaks for which various cost functions could be tested.

As far as we are aware, every previous model for cost constrained genome rearrangements was *symmetric*. By this we mean that for every rearrangement scenario between genomes A and B, there exist scenarios transforming A into B, and B into A of the same cost. Such a model does not provide any information regarding the branch of a phylogenetic tree on which rearrangements occurred. Our models, on the other hand, can be non-symmetric. In this case it might happen that the black sure 2-break has *significantly* lower cost than the gray one for the same circle subgraph of the genome breakpoint graph. If the true evolutionary scenario is *close* to being parsimonious and the cost function is biologically relevant, then this indicates that a particular rearrangement happened on the lineage leading to genome A. In addition to this, in our work a DCJ scenario transforms genomes A and B into some genome C. For the true evolutionary scenario this C would be the least common ancestor of A and B. If a biologically relevant cost function φ is available, then the genome C thus obtained for a φ-MCPS scenario might inform us about the properties of the least common ancestor of A and B. If we were able to enumerate or uniformly sample the φ-MCPS scenarios, then we could ask various questions concerning the gene order of the least common ancestor and the properties of its intergenic regions.

Chapter 7

Résumé

Les Réarrangements Génomiques Dans le Contexte Évolutif

Un *réarrangement génomique* est une mutation qui modifie la structure des chromosomes voire même leur nombre dans un génome. Prenez par exemple les humains et les chimpanzés. Nous avons 23 paires de chromosomes, alors que les chimpanzés en ont 24. Cette différence est due à une fusion de deux chromosomes ancestraux qui a donné naissance au chromosome 2 chez l'homme [37]. Les lignées humains et chimpanzés ont chacune accumulé un certain nombre d'autres réarrangements depuis leur séparation [65].

Outre des *fusions* et des *fissions* de chromosomes, ces réarrangements comprennent des *délétions*, des *insertions* et des *inversions* de segments chromosomiques. Deux extrémités de chromosomes différents peuvent également être échangées au cours d'une *translocation*. L'ensemble de ces mutations constitue un *scénario évolutif de réarrangements* entre les espèces. Nous nous sommes intéressés à la reconstruction des scénarios de réarrangements entre espèces animales.

Les progrès récents dans les technologies de séquençage nous fournissent une occasion sans précédent pour étudier les mécanismes moléculaires de réarrangements génomiques [71, 66], la façon dont ils se propagent dans les populations [40], et leur importance pour l'évolution [74]. Des méthodes de détection systématique de réarrangements génomiques dans le génome humain émergent [60] et nous informent de leur apparition en chacun de nous. Des études récentes s'appuient sur ces avancées pour dévoiler le rôle joué dans les scénarios évolutifs par:

- les domaines de chromatine transcriptionnellement actifs [50, 53],
- la proximité spatiale entre loci dans un génome [99, 17, 94],

- les limites des domaines d'association topologique [68, 50, 61],
- les structures d'ADN non canoniques [52].

Notre projet associe des outils mathématiques et algorithmiques avec la compréhension biologique actuelle des réarrangements génomiques. D'un point de vue biologique, notre objectif est de lier génétique et épigénétique aux réarrangements dans les deux sens :

- nous développons une méthodologie pour étudier des caractéristiques génétiques et épigénétiques associées aux réarrangements, et inversement
- pour trouver des scénarios de réarrangements guidés par de telles caractéristiques génétiques et épigénétiques.

Des Modèles Mathématiques de Réarrangements Pondérés

Les études mathématiques et algorithmiques des scénarios de réarrangements génomiques ont commencé il y a un quart de siècle et ont conduit à un grand nombre de problèmes combinatoires [48]. Le problème central dans ce domaine est celui de trouver un scénario optimal transformant un génome en un autre.

Les réarrangements ont généralement été traités comme étant tout aussi probables, ou ayant des poids égaux. Dans ce cas, un scénario est optimal s'il est de longueur minimale. Nous présentons dans la Section 1.3 un aperçu des modèles dans lesquels des poids différents ont été attribués aux réarrangements en fonction de certaines contraintes biologiques, dans le but de trouver des scénarios de réarrangement qui ressemblent mieux au véritable scénario évolutif. Cependant, dans la plupart des cas, ces contraintes supplémentaires entraînent des problèmes difficiles. Des algorithmes polynomiaux sont connus seulement pour quelques-uns d'entre eux:

- Un scénario le plus court composé d'inversions affectant au plus deux gènes peut être trouvé en temps polynomial, comme le montrent Galvao, Baudet, et Dias [51] et Bender, Ge, He, Hu, Pinter, Skiena, et Swidan [10].
- Si le poids d'inversion est égal au nombre de gènes qu'il affecte, alors un scénario de poids minimum composé d'inversions peut être trouvé en temps polynomial pour deux chaînes binaires [10].
- Un scénario *parfait* de longueur minimale composé d'inversions peut être trouvé en temps polynomial pour une certaine famille d'intervalles communs, comme le montre Bérard, Bergeron, Chauve, et Paul [11].

À notre connaissance, il existe d'autres algorithmes polynomiaux pour les problèmes de réarrangements pondérés : ce sont ceux qui s'appuient sur un modèle de réarrangements appelés *double cut and join*. Le double cut and join (DCJ) a été introduit en 2005 par Yancopoulos, Attie et Friedberg [102], et est mathématiquement beaucoup plus simple que les modèles précédents basés sur les inversions ou les inversions et les translocations. Le double cut and join modélise un réarrangement en coupant des chromosomes en un ou deux endroits et en rejoignant les brins chromosomiques. En plus des inversions et des translocations, cette opération simple peut reproduire d'autres réarrangements comme une circularisation d'un chromosome linéaire et une excision d'un chromosome circulaire. On ne sait pas si ces derniers types de réarrangements ont un rôle dans l'évolution, mais des éléments d'ADN circulaires ont été trouvés dans des cellules humaines à l'état normal [76] et sont abondantes dans des cellules cancéreuses [98]. Il existe quelques algorithmes efficaces pour les problèmes de double cut and join pondérés :

- Un scénario *parfait* de longueur minimale de double cut and join peut être trouvé en temps polynomial pour une certaine famille d'intervalles communs, comme le montrent Bérard, Chateau, Chauve, Paul, and Tannier [12].
- Un scénario de poids minimum peut être trouvé en temps polynomial parmi les scénarios de longueur minimale pour les double cut and join pondérés en fonction des insertions et des suppressions dans les régions intergéniques, comme le montrent Bulteau, Fertin et Tannier [27].
- Un scénario de poids minimum peut être trouvé en temps polynomial parmi les scénarios de longueur minimale pour les double cut and join pondérés en fonction d'emplacements spatiaux des régions intergéniques, comme nous l'avons montré dans nos travaux précédents [95, 90, 89],

Nous soulignons dans la Section 1.4 que les données biologiques qui sont disponibles aujourd'hui nécessitent des modèles de réarrangements pondérés qui sont beaucoup plus flexibles que ceux qui ont été étudiés auparavant.

Un Cadre Général pour les Double Cut and Join Pondérés

La principale contribution de cette thèse est la suivante :

Projet. *Nous présentons un cadre sur le modèle de réarrangements double cut and join avec des poids arbitraires. Dans ce cadre un scénario de poids minimum peut être trouvé en temps polynomial parmi les scénarios de longueur minimale pour deux génomes à contenu génétique identique et sans doublons.*

Cela signifie que notre objectif n'est pas de proposer un modèle particulier de pondération de double cut and join d'une manière biologiquement significative, mais plutôt de guider ce processus et de garantir qu'aucun travail algorithmique n'est nécessaire si nos directives sont respectées.

Les axes principaux de notre projet sont les suivants:

1. Une méthode pour explorer efficacement l'espace des scénarios double cut and join de longueur minimale.
2. Une méthode pour enrichir un réarrangement avec des informations concernant les régions intergéniques qu'il coupe et rejoint, et un poids arbitraire.
3. Un algorithme efficace pour trouver un scénario de poids minimum parmi les scénarios de longueur minimale.

Ces tâches sont traitées respectivement aux Chapitre 3, Chapitre 4, et Chapitre 5. Notre cadre généralise les trois modèles de double cut and join pondérés précedemment évoqués [95, 27, 12].

Aperçu

Au Chapitre 2, nous montrons qu'un double cut and join peut être interprété comme une transformation de graphe. Dans ce qui suit nous introduisons en fait un cadre de transformations pondérées des graphes et non de réarrangements pondérées génomiques.

Un *2-break* est une transformation de graphe échangeant les extrémités de deux arêtes comme illustré en Figure 7.1. Nous montrons en Section 2.7 qu'un 2-break généralise un certain nombre d'opérations de tri. Celles-ci incluent le tri des permutations avec transpositions [41], le tri des chaînes avec échanges [4], le tri des génomes avec double cut and join [102], et l'échange des jetons sur les graphes [23]. Des variantes pondérées pour tous ces problèmes ont été étudiées dans la littérature. Cela a motivé notre choix d'étudier les 2-breaks pondérés et pas seulement le cas particulier du double cut and join.

Au Chapitre 3, nous examinons un scénario de 2-breaks de longueur minimale pour un graphe et démontrons qu'il peut être partitionné en scénarios de 2-breaks de longueur minimale pour des cycles. Nous procédons ensuite en montrant qu'un scénario de 2-breaks de longueur minimale pour un cycle peut être partitionné en scénarios pour des cycles qui sont plus petits. Cela conduit à un algorithme de programmation dynamique qui explore l'espace des scénarios de 2-breaks de longueur minimale pour un cycle.

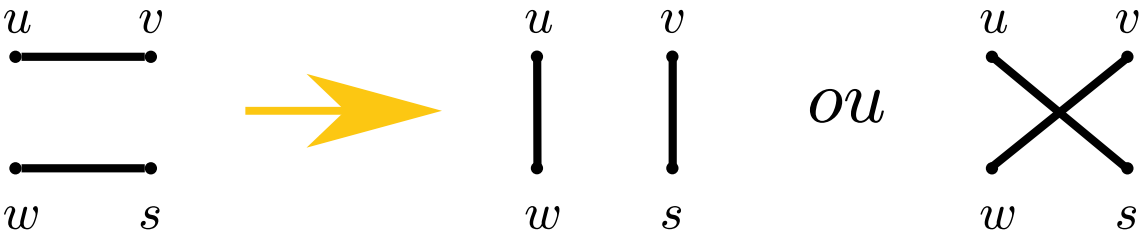


Figure 7.1: Un 2-break transforme une paire d'arêtes $\{u, v\}$ et $\{w, s\}$ en $\{u, w\}$ et $\{v, s\}$, ou $\{u, s\}$ et $\{v, w\}$.

Au Chapitre 4, nous présentons notre cadre de 2-breaks pondérés et expliquons qu'il généralise les travaux antérieurs.

Au Chapitre 5, nous fournissons un algorithme exact pour trouver un scénario de poids minimum parmi les scénarios de 2-breaks de longueur minimale. En l'utilisant, nous pouvons trouver, en temps $O(n^5L^4)$, un scénario de poids minimum parmi les scénarios de réarrangements pondérés de longueur minimale pour deux génomes à contenu génétique identique et sans doublons. Ici n est le nombre de gènes et L est le nombre d'états différents auxquels une région intergénique est autorisée à accéder. Le problème devient NP-difficile si les génomes ne satisfont pas à ces critères. Dans ce cas, nous utilisons l'optimisation linéaire en nombres entiers pour résoudre le problème de manière exacte. Nous concluons le chapitre par une discussion concernant la complexité polynomial de notre algorithme. Nous montrons qu'en pratique n est assez petit, comme on l'a déjà discuté dans [89].

Le vrai scénario évolutif pourrait être d'une longueur non minimale. Il serait donc intéressant d'explorer l'espace de tous les scénarios, et pas seulement ceux de longueur minimale. Cependant on sait que c'est NP-difficile de trouver un scénario de réarrangements pondérés de poids minimum, même pour les modèles très simples de pondération, comme le montrent Fertin, Jean, and Tannier [27] et nos travaux [91].

Nous estimons qu'il n'est pas nécessaire d'explorer l'espace de tous les scénarios possibles. D'abord, des outils statistiques pourraient être utilisés pour estimer une borne supérieure l pour la longueur du véritable scénario évolutif, comme discuté par Biller, Guéguen, Knibbe, et Tannier [20] et Alexeev et Alekseyev [2]. Ensuite, seuls les scénarios de longueur inférieure à l pourraient être explorés. Un certain nombre de nos résultats sont en fait prouvés pour les scénarios qui ne sont pas nécessairement de longueur minimale. Ces résultats fournissent donc une base pour les travaux futurs.

Au Chapitre 6 nous discutons comment la pertinence d'un modèle de réarrangements pondérés pourrait être testée en utilisant des données biologiques. Pour conclure, nous expliquons comment notre cadre pourrait être utile pour les études phylogénétiques.

Bibliography

[1] Max A Alekseyev and Pavel A Pevzner. Breakpoint graphs and ancestral genome reconstructions. *Genome Research*, 19(5):943–957, 2009.

[2] Nikita Alexeev and Max A Alekseyev. Estimation of the true evolutionary distance under the fragile breakage model. *BMC Genomics*, 18(4):356, 2017.

[3] Amihood Amir, Tzvika Hartman, Oren Kapah, Avivit Levy, and Ely Porat. On the cost of interchange rearrangement in strings. *SIAM Journal on Computing*, 39(4):1444–1461, 2010.

[4] Amihood Amir and Avivit Levy. String rearrangement metrics: A survey. *Algorithms and Applications*, 2010.

[5] Nikos Apostolakis. Non-crossing trees, quadrangular dissections, ternary trees, and duality preserving bijections. *arXiv Preprint arXiv:1807.11602*, 2018.

[6] Christian Bachmaier, Franz J Brandenburg, Andreas Gleißner, and Andreas Hofmeier. On the hardness of maximum rank aggregation problems. *Journal of Discrete Algorithms*, 31:2–13, 2015.

[7] Vineet Bafna and Pavel A Pevzner. Genome rearrangements and sorting by reversals. *SIAM Journal on Computing*, 25(2):272–289, 1996.

[8] Yuliy Baryshnikov. On stokes sets. In *New Developments in Singularity Theory*, pages 65–86. Springer, 2001.

[9] Christian Baudet, Ulisses Dias, and Zanoni Dias. Sorting by weighted inversions considering length and symmetry. *BMC Bioinformatics*, 16(S19):S3, 2015.

[10] Michael A Bender, Dongdong Ge, Simai He, Haodong Hu, Ron Y Pinter, Steven Skiena, and Firas Swidan. Improved bounds on sorting by length-weighted reversals. *Journal of Computer and System Sciences*, 74(5):744–774, 2008.

[11] Severine Bérard, Anne Bergeron, Cedric Chauve, and Christophe Paul. Perfect sorting by reversals is not always difficult. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(1):4–16, 2007.

[12] Sèverine Bérard, Annie Chateau, Cedric Chauve, Christophe Paul, and Eric Tannier. Computation of perfect DCJ rearrangement scenarios with linear and circular chromosomes. *Journal of Computational Biology*, 16(10):1287–1309, 2009.

[13] Anne Bergeron. A very elementary presentation of the Hannenhalli-Pevzner theory. In *Annual Symposium on Combinatorial Pattern Matching*, pages 106–117. Springer, 2001.

[14] Anne Bergeron, Cedric Chauve, Tzvika Hartman, and Karine St-Onge. On the properties of sequences of reversals that sort a signed permutation. In *Proceedings of JOBIM*, volume 2, pages 99–108. Citeseer, 2002.

[15] Anne Bergeron, Julia Mixtacki, and Jens Stoye. A unifying view of genome rearrangements. In *International Workshop on Algorithms in Bioinformatics*, pages 163–173. Springer, 2006.

[16] Matthias Bernt, Anke Braband, Bernd Schierwater, and Peter F Stadler. Genetic aspects of mitochondrial genome evolution. *Molecular Phylogenetics and Evolution*, 69(2):328–338, 2013.

[17] Camille Berthelot, Matthieu Muffato, Judith Abecassis, and Hugues Roest Crollius. The 3D organization of chromatin explains evolutionary fragile genomic regions. *Cell Reports*, 10(11):1913–1924, 2015.

[18] Sangeeta Bhatia, Pedro Feijão, and Andrew R Francis. Position and content paradigms in genome rearrangements: the wild and crazy world of permutations in genomics. *Bulletin of Mathematical Biology*, 80(12):3227–3246, 2018.

[19] Daniel Bienstock and Oktay Günlük. A degree sequence problem related to network design. *Networks*, 24(4):195–205, 1994.

[20] Priscila Biller, Laurent Guéguen, Carole Knibbe, and Eric Tannier. Breaking good: accounting for fragility of genomic regions in rearrangement distance estimation. *Genome Biology and Evolution*, 8(5):1427–1439, 2016.

[21] James R Bitner. An asymptotically optimal algorithm for the Dutch National Flag problem. *SIAM Journal on Computing*, 11(2):243–262, 1982.

[22] Mathieu Blanchette, Takashi Kunisawa, and David Sankoff. Parametric genome rearrangement. *Gene*, 172(1):GC11–GC17, 1996.

[23] Édouard Bonnet, Tillmann Miltzow, and Paweł Rzążewski. Complexity of token swapping and its variants. *Algorithmica*, 80(9):2656–2682, 2018.

[24] Marília DV Braga and Jens Stoye. Counting all DCJ sorting scenarios. In *RECOMB International Workshop on Comparative Genomics*, pages 36–47. Springer, 2009.

[25] Marília DV Braga and Jens Stoye. The solution space of sorting by DCJ. *Journal of Computational Biology*, 17(9):1145–1165, 2010.

[26] Klairton Lima Brito, Géraldine Jean, Guillaume Fertin, Andre Rodrigues Oliveira, Ulisses Dias, and Zanoni Dias. Sorting by genome rearrangements on both gene order and intergenic sizes. *Journal of Computational Biology*, 27(2):156–174, 2020.

[27] Laurent Bulteau, Guillaume Fertin, and Eric Tannier. Genome rearrangements with indels in intergenes restrict the scenario space. *BMC Bioinformatics*, 17(14):426, 2016.

[28] Alberto Caprara. Sorting permutations by reversals and Eulerian cycle decompositions. *SIAM Journal on Discrete Mathematics*, 12(1):91–110, 1999.

[29] Alberto Caprara. The reversal median problem. *INFORMS Journal on Computing*, 15(1):93–113, 2003.

[30] Arthur Cayley. LXXVII. Note on the theory of permutations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 34(232):527–529, 1849.

[31] Xin Chen. On sorting unsigned permutations by double-cut-and-joins. *Journal of Combinatorial Optimization*, 25(3):339–351, 2013.

[32] Leonid Chindelevitch, Sean La, and Joao Meidanis. A cubic algorithm for the generalized rank median of three genomes. *Algorithms for Molecular Biology*, 14(1):16, 2019.

[33] Leonid Chindelevitch, João Paulo Pereira Zanetti, and João Meidanis. On the rank-distance median of 3 permutations. *BMC Bioinformatics*, 19(6):142, 2018.

[34] Luís Felipe I Cunha, Pedro Feijão, Vinícius F dos Santos, Luis Antonio B Kowada, and Celina MH de Figueiredo. On the computational complexity of closest genome problems. *Discrete Applied Mathematics*, 274:26–34, 2020.

[35] Aaron E Darling, István Miklós, and Mark A Ragan. Dynamics of genome rearrangement in bacterial populations. *PLoS Genetics*, 4(7), 2008.

[36] Jean De Grouchy. Chromosome phylogenies of man, great apes, and old world monkeys. *Genetica*, 73(1-2):37–52, 1987.

[37] Timothy R Dreszer, Gregory D Wall, David Haussler, and Katherine S Pollard. Biased clustered substitutions in the human genome: the footprints of male-driven biased gene conversion. *Genome Research*, 17(10):1420–1430, 2007.

[38] Serge Dulucq and Jean-Guy Penaud. Cordes, arbres et permutations. *Discrete Mathematics*, 117(1-3):89–105, 1993.

[39] Jonathan A Eisen, John F Heidelberg, Owen White, and Steven L Salzberg. Evidence for symmetric chromosomal inversions around the replication origin in bacteria. *Genome Biology*, 1(6):1–9, 2000.

[40] Rui Faria, Kerstin Johannesson, Roger K Butlin, and Anja M Westram. Evolving inversions. *Trends in Ecology & Evolution*, 2019.

[41] Farzad Farnoud and Olgica Milenkovic. Sorting of permutations by cost-constrained transpositions. *IEEE Transactions on Information Theory*, 58(1):3–23, 2012.

[42] Farzad Farnoud, Olgica Milenkovic, Gregory J Puleo, and Lili Su. Computing similarity distances between rankings. *Discrete Applied Mathematics*, 232:157–175, 2017.

[43] Tomás Feder, Adam Guetz, Milena Mihail, and Amin Saberi. A local switch Markov chain on given degree graphs with application in connectivity of peer-to-peer networks. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 69–76. IEEE, 2006.

[44] Pedro Feijão and Eloi Araujo. Fast ancestral gene order reconstruction of genomes with unequal gene content. *BMC Bioinformatics*, 17(14):413, 2016.

[45] Pedro Feijão and Joao Meidanis. SCJ: a breakpoint-like distance that simplifies several rearrangement problems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(5):1318–1329, 2011.

[46] Pedro Feijao and Joao Meidanis. Extending the algebraic formalism for genome rearrangements to include linear chromosomes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(4):819–831, 2012.

[47] Guillaume Fertin, Géraldine Jean, and Eric Tannier. Algorithms for computing the double cut and join distance on both gene order and intergenic sizes. *Algorithms for Molecular Biology*, 12(1):16, 2017.

[48] Guillaume Fertin, Anthony Labarre, Irena Rusu, Stéphane Vialette, and Eric Tannier. *Combinatorics of genome rearrangements*. MIT press, 2009.

[49] Bailey K Fosdick, Daniel B Larremore, Joel Nishimura, and Johan Ugander. Configuring random graph models with fixed degree sequences. *SIAM Review*, 60(2):315–355, 2018.

[50] Geoff Fudenberg and Katherine S Pollard. Chromatin features constrain structural variation across evolutionary timescales. *Proceedings of the National Academy of Sciences*, 116(6):2175–2180, 2019.

[51] Gustavo Rodrigues Galvao, Christian Baudet, and Zanoni Dias. Sorting circular permutations by super short reversals. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(3):620–633, 2016.

[52] Ilias Georgakopoulos-Soares, Sandro Morganella, Naman Jain, Martin Hemberg, and Serena Nik-Zainal. Noncanonical secondary structures arising from non-B DNA motifs are determinants of mutagenesis. *Genome Research*, 28(9):1264–1271, 2018.

[53] Henrike Johanna Gothe, Britta Annika Maria Bouwman, Eduardo Gade Gusmao, Rossana Piccinno, Giuseppe Petrosino, Sergi Sayols, Oliver Drechsel, Vera Minneker, Natasa Josipovic, Athanasia Mizi, et al. Spatial chromosome folding and active transcription drive DNA fragility and formation of oncogenic MLL translocations. *Molecular Cell*, 75(2):267–283, 2019.

[54] Sridhar Hannenhalli and Pavel A Pevzner. Transforming men into mice (polynomial algorithm for genomic distance problem). In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 581–592. IEEE, 1995.

[55] Sridhar Hannenhalli and Pavel A Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM (JACM)*, 46(1):1–27, 1999.

[56] Tom Hartmann, Matthias Bernt, and Martin Middendorf. An exact algorithm for sorting by weighted preserving genome rearrangements. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(1):52–62, 2018.

[57] Tom Hartmann, Martin Middendorf, and Matthias Bernt. Genome rearrangement analysis: Cut and join genome rearrangements and gene cluster preserving approaches. In *Comparative Genomics*, pages 261–289. Springer, 2018.

[58] Tom Hartmann, Nicolas Wieseke, Roded Sharan, Martin Middendorf, and Matthias Bernt. Genome rearrangement with ILP. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(5):1585–1593, 2017.

[59] Steffen Heber and Jens Stoye. Algorithms for finding gene clusters. In *International Workshop on Algorithms in Bioinformatics*, pages 252–263. Springer, 2001.

[60] Steve S Ho, Alexander E Urban, and Ryan E Mills. Structural variation in the sequencing era. *Nature Reviews Genetics*, pages 1–19, 2019.

[61] Linh Huynh and Fereydoun Hormozdiari. TAD fusion score: discovery and ranking the contribution of deletions to genome structure. *Genome Biology*, 20(1):60, 2019.

[62] Ekhine Irurozki, Borja Calvo, and Jose A Lozano. Sampling and learning Mallows and Generalized Mallows models under the Cayley distance. *Methodology and Computing in Applied Probability*, 20(1):1–35, 2018.

[63] Donald B Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM (JACM)*, 24(1):1–13, 1977.

[64] Oren Kapah, Gad M Landau, Avivit Levy, and Nitsan Oz. Interchange rearrangement: The element-cost model. *Theoretical Computer Science*, 410(43):4315–4326, 2009.

[65] Jaebum Kim, Marta Farré, Loretta Auvil, Boris Capitanu, Denis M Larkin, Jian Ma, and Harris A Lewin. Reconstruction and evolutionary history of eutherian chromosomes. *Proceedings of the National Academy of Sciences*, 114(27):E5379–E5388, 2017.

[66] Seoyoung Kim, Shaun E Peterson, Maria Jasin, and Scott Keeney. Mechanisms of germ line genome instability. In *Seminars in Cell & Developmental Biology*, volume 54, pages 177–187. Elsevier, 2016.

[67] GT Klincsek. Minimal triangulations of polygonal domains. *Ann. Discrete Math*, 9:121–123, 1980.

[68] Nathan H Lazar, Kimberly A Nevonen, Brendan O'Connell, Christine McCann, Rachel J O'Neill, Richard E Green, Thomas J Meyer, Mariam Okhovat, and Lucia Carbone. Epigenetic maintenance of topological domains in the highly rearranged gibbon genome. *Genome Research*, 28(7):983–997, 2018.

[69] Yu Lin and Bernard ME Moret. Estimating true evolutionary distances under the DCJ model. *Bioinformatics*, 24(13):i114–i122, 2008.

[70] Carla Negri Lintzmayer, Guillaume Fertin, and Zanoni Dias. Sorting permutations and binary strings by length-weighted rearrangements. *Theoretical Computer Science*, 715:35–59, 2018.

[71] Ram-Shankar Mani and Arul M Chinnaiyan. Triggers for genomic rearrangements: insights into genomic, cellular and environmental influences. *Nature Reviews Genetics*, 11(12):819–829, 2010.

[72] Fady Massarwi, Boris van Sosin, and Gershon Elber. Untrimming: Precise conversion of trimmed-surfaces to tensor-product surfaces. *Computers & Graphics*, 70:80–91, 2018.

[73] João Meidanis and Zanoni Dias. An alternative algebraic formalism for genome rearrangements. In *Comparative Genomics*, pages 213–223. Springer, 2000.

[74] Claire Mérot, Rebekah A Oomen, Anna Tigano, and Maren Wellenreuther. A roadmap for understanding the evolutionary significance of structural genomic variation. *Trends in Ecology & Evolution*, 2020.

[75] István Miklós and Heather Smith. Sampling and counting genome rearrangement scenarios. *BMC Bioinformatics*, 16(14):S6, 2015.

[76] Henrik Devitt Møller, Marghoob Mohiyuddin, Iñigo Prada-Luengo, M Reza Sailani, Jens Frey Halling, Peter Plomgaard, Lasse Maretty, Anders Johannes Hansen, Michael P Snyder, Henriette Pilegaard, et al. Circular DNA elements of chromosomal origin are common in healthy human somatic tissue. *Nature Communications*, 9(1):1–12, 2018.

[77] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118, 2001.

[78] Enno Ohlebusch, Mohamed I Abouelhoda, and Kathrin Hockel. A linear time algorithm for the inversion median problem in circular bacterial genomes. *Journal of Discrete Algorithms*, 5(4):637–646, 2007.

[79] Andre R Oliveira, Géraldine Jean, Guillaume Fertin, Ulisses Dias, and Zanoni Dias. Super short operations on both gene order and intergenic sizes. *Algorithms for Molecular Biology*, 14(1):1–17, 2019.

[80] Andre Rodrigues Oliveira, Klairton Lima Brito, Ulisses Dias, and Zanoni Dias. On the complexity of sorting by reversals and transpositions problems. *Journal of Computational Biology*, 26(11):1223–1229, 2019.

[81] Aïda Ouangraoua and Anne Bergeron. Combinatorial structure of genome rearrangements scenarios. *Journal of Computational Biology*, 17(9):1129–1144, 2010.

[82] V Yu Popov. Multiple genome rearrangement by swaps and by element duplications. *Theoretical Computer Science*, 385(1-3):115–126, 2007.

[83] José M Ranz, Damien Maurin, Yuk S Chan, Marcin Von Grotthuss, LaDeana W Hillier, John Roote, Michael Ashburner, and Casey M Bergman. Principles of genome evolution in the Drosophila melanogaster species group. *PLoS Biology*, 5(6), 2007.

[84] David Sankoff. Edit distance for genome comparison based on non-local operations. In *Annual Symposium on Combinatorial Pattern Matching*, pages 121–135. Springer, 1992.

[85] Cathal Seoighe, Nancy Federspiel, Ted Jones, Nancy Hansen, Vesna Bivolarovic, Ray Surzycki, Raquel Tamse, Caridad Komp, Lucas Huizar, Ronald W Davis, et al. Prevalence of small inversions in yeast gene order evolution. *Proceedings of the National Academy of Sciences*, 97(26):14433–14437, 2000.

[86] Mingfu Shao and Yu Lin. Approximating the edit distance for genomes with duplicate genes under DCJ, insertion and deletion. In *BMC Bioinformatics*, volume 13, page S13. Springer, 2012.

[87] Mingfu Shao, Yu Lin, and Bernard Moret. Sorting genomes with rearrangements and segmental duplications through trajectory graphs. In *BMC Bioinformatics*, volume 14, page S9. Springer, 2013.

[88] Mingfu Shao, Yu Lin, and Bernard ME Moret. An exact algorithm to compute the double-cut-and-join distance for genomes with duplicate genes. *Journal of Computational Biology*, 22(5):425–435, 2015.

[89] Pijus Simonaitis, Annie Chateau, and Krister Swenson. Weighted minimum-length rearrangement scenarios. In *19th International Workshop on Algorithms in Bioinformatics (WABI)*, pages 13–1, 2019.

[90] Pijus Simonaitis, Annie Chateau, and Krister M Swenson. A general framework for genome rearrangement with biological constraints. *Algorithms for Molecular Biology*, 14(1):15, 2019.

[91] Pijus Simonaitis and Krister M Swenson. Finding local genome rearrangements. *Algorithms for Molecular Biology*, 13(1):9, 2018.

[92] Daniela C Soto, Colin Shew, Mira Mastoras, Joshua M Schmidt, Ruta Sahasrabudhe, Gulhan Kaya, Aida M Andrés, and Megan Y Dennis. Identification of structural variation in chimpanzees using optical mapping and nanopore sequencing. *Genes*, 11(3):276, 2020.

[93] AH Sturtevant. A case of rearrangement of genes in drosophila. *Proceedings of the National Academy of Sciences of the United States of America*, 7(8):235, 1921.

[94] Krister M Swenson and Mathieu Blanchette. Large-scale mammalian genome rearrangements coincide with chromatin interactions. *Bioinformatics*, 35(14):i117–i126, 2019.

[95] Krister M Swenson, Pijus Simonaitis, and Mathieu Blanchette. Models and algorithms for genome rearrangement with positional constraints. *Algorithms for Molecular Biology*, 11(1):13, 2016.

[96] Javier Tamames, Georg Casari, Christos Ouzounis, and Alfonso Valencia. Conserved clusters of functionally related genes in two bacterial genomes. *Journal of Molecular Evolution*, 44(1):66–73, 1997.

[97] Eric Tannier, Chunfang Zheng, and David Sankoff. Multichromosomal median and halving problems under different genomic distances. *BMC Bioinformatics*, 10(1):120, 2009.

[98] Kristen M Turner, Viraj Deshpande, Doruk Beyter, Tomoyuki Koga, Jessica Rusert, Catherine Lee, Bin Li, Karen Arden, Bing Ren, David A Nathanson, et al. Extrachromosomal oncogene amplification drives tumour evolution and genetic heterogeneity. *Nature*, 543(7643):122–125, 2017.

[99] Amélie S Véron, Claire Lemaitre, Christian Gautier, Vincent Lacroix, and Marie-France Sagot. Close 3D proximity of evolutionary breakpoints argues for the notion of spatial synteny. *BMC Genomics*, 12(1):303, 2011.

[100] Katsuhisa Yamanaka, Erik D Demaine, Takehiro Ito, Jun Kawahara, Masashi Kiyomi, Yoshio Okamoto, Toshiki Saitoh, Akira Suzuki, Kei Uchizawa, and Takeaki Uno. Swapping labeled tokens on graphs. *Theoretical Computer Science*, 586:81–94, 2015.

[101] Katsuhisa Yamanaka, Takashi Horiyama, J Mark Keil, David Kirkpatrick, Yota Otachi, Toshiki Saitoh, Ryuhei Uehara, and Yushi Uno. Swapping colored tokens on graphs. *Theoretical Computer Science*, 729:1–10, 2018.

[102] Sophia Yancopoulos, Oliver Attie, and Richard Friedberg. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics*, 21(16):3340–3346, 2005.

[103] Sophia Yancopoulos and Richard Friedberg. Sorting genomes with insertions, deletions and duplications by DCJ. In *RECOMB International Workshop on Comparative Genomics*, pages 170–183. Springer, 2008.

[104] Siyi Yang, Clayton Schoeny, and Lara Dolecek. Theoretical bounds and constructions of codes in the generalized Cayley metric. *IEEE Transactions on Information Theory*, 65(8):4746–4763, 2019.

[105] Joao Paulo Pereira Zanetti, Priscila Biller, and Joao Meidanis. Median approximations for genomes modeled as matrices. *Bulletin of Mathematical Biology*, 78(4):786–814, 2016.