Technische Universität Berlin
Electrical Engineering and Computer Science
Institute of Software Engineering and Theoretical Computer Science
Algorithmics and Computational Complexity (AKT)

On Reachable Assignments in Social Networks

Bachelorarbeit

von Luis Müller

zur Erlangung des Grades „Bachelor of Science“ (B. Sc.)
im Studiengang Informatik

Erstgutachter:	Prof. Dr. Rolf Niedermeier
Zweitgutachter:	Prof. Dr. Markus Brill
Betreuer:	Matthias Bentert, Prof. Dr. Rolf Niedermeier

Zusammenfassung

Ein häufiges Problem im Feld der Multiagentensysteme ist die effiziente Verteilung und Zuweisung von Resourcen unter einer Gruppe von Agenten. Diese Resourcen werden oft als teilbare bzw. unteilbare Güter oder Objekte modelliert. In dieser Arbeit untersuchen wir das Reachable Assignment Problem, welches von Gourves, Lesca únd Wilczynski [AAAI, 2017] eingeführt wurde. Gegeben eine Menge von Agenten, sowie eine Menge von Objekten, eine initialen Verteilung dieser Objekte, sodass jeder Agent genau ein Objekt besitzt, die Objektpräferenzen der Agenten und einen Graphen, der kodiert, welche Agenten Resourcen mit welchen anderen Agenten tauschen können, ist die Fragestellung des Reachable Assignment Problems, ob eine bestimmte Zielzuweisung über eine Sequenz von rationalen Tauschen erreichbar ist. Ein rationaler Tausch ist ein Tausch, bei dem beide Tauschpartner eine Resource erhalten, die sie über die vorherige Resource präferieren. Bei der Untersuchung von Kreisen entwickeln wir einen Algorithmus, welcher Reachable Assignmentauf Kreisen in $\mathcal{O}(n^3)$ Zeit löst. Desweiteren untersuchen wir Cliquen und zeigen, dass Reachable Assignment in diesem Fall NP-hart ist. Dies zeigen wir durch eine Reduktion vom Reachable Object Problem, welches wie Reachable Assignment konstruiert ist, jedoch hierbei die Frage ist, ob ein bestimmter Agent durch eine Sequenz von rationalen Tauschen ein bestimmtes Objekt erhalten kann.

Abstract

A frequent problem in the field of *Multi-Agent-Systems* is the efficient distribution and allocation of resources among agents. These resources are often modelled as divisible or indivisible goods or objects. In this work, we will investigate the Reachable Assignment problem, introduced by Gourves, Lesca and Wilczynski [AAAI, 2017]. Given a set of agents, a set of objects, an initial distribution of the objects, where each agent owns exactly one object, knowledge of the agent's preferences and a graph encoding which agents can trade resources with which others, the question is whether a certain target assignment is reachable via a sequence of rational trades. A rational trade is a trade after which both involved agents possess an object they prefer over the object they held before the trade. Investigating graphs that are cycles, we develop an algorithm to solve Reachable Assignment on cycles in $\mathcal{O}(n^3)$ time, where n is the number of agents involved. Further we investigate into graphs that are cliques and show that Reachable Assignment on cliques in NP-hard via a reduction from Reachable Object, which is constructed akin to Reachable Assignment, except the question is whether a designated agent can obtain a designated object via a sequence of rational trades among the agents.

Contents

1 Introduction

A frequent problem in *Multi-Agent-Systems* is the efficient distribution of resources among agents [Che+06]. Sometimes these resources appear as bundles, sometimes they are indivisible objects. The field which studies procedures to allocate divisible or indivisible resources to agents is called *Resource Allocation*. One famous problem in this field is called the *house marketing problem* where each of the n participating agents initially owns one house and is able to trade these houses with other agents. House marketing is also known in the field of economics [Sha74]. The house marketing problem is an example of a resource allocation problem with indivisible goods. Gourves et al. [GLW17] study a generalization of the house marketing problem where agents are only able to perform a trade of goods with agents they trust, which can be modelled in a social network of participating agents where an edge between two agents means that they trust each other.

We will study the case where the social network is a cycle, i.e. each agent has exactly two other agents it can trade goods with. Afterwards we will return to a problem even closer to the house marketing problem where each agent can swap resources with every other agent.

1.1 A Generalization of the House Marketing Problem: Reachability of Resource Allocations

Let N be a set of n agents and let X be a set of n indivisible objects. Each agent $i \in N$ has a preference list $\succ_i$ over a non-empty subset X_i of the set of objects X. Each preference list is a strict ordering on X_i. The set of all preference lists is called a preference profile $\succ$ [GLW17]. A bijection $\sigma : N \to X$ is called an allocation or assignment. Akin to the house marketing problem, each agent is initially assigned exactly one object. We denote this initial assignment by σ_0. Let G be a graph where there is a bijection between the set of vertices $V(G)$ and the set of agents N. We will use the term agents interchangeably with the vertices of G.

We say that a trade between agents i and j is only possible if their corresponding vertices share an edge in G and if both i and j receive an object that they prefer to their currently assigned object. We can express this formally as

$$\sigma(i) \succ_j \sigma(j) \tag{1.1}$$

and

$$\sigma(j) \succ_i \sigma(i). \tag{1.2}$$

We call such a trade a rational trade or a *swap*. A *swap sequence* is a sequence of assignments $(\sigma_s, ..., \sigma_t)$ where σ_i is the result of performing one swap in assignment σ_{i-1} for all $i \in \{s+1, s+2, ..., t\}$. We call an assignment σ reachable if there exists a sequence of swaps $(\sigma_0, ..., \sigma)$. For an agent i, an object x is reachable if there exists an assignment σ_t and a swap sequence $(\sigma_0, ..., \sigma_t)$ such that $\sigma_t(i) = x$. We will now state the problem descriptions of Gourves et al. [GLW17] for Reachable Assignment and Reachable Object.

Reachable Object
Input: A set of agents N, a set of objects X, a preference profile $\succ$, a graph G, an initial assignment σ_0, an agent i and an object x.
Question: Is x reachable for i from σ_0?

Reachable Assignment
Input: A set of agents N, a set of objects X, a preference profile $\succ$, a graph G, an initial assignment σ_0 and a target assignment σ.
Question: Is σ reachable from σ_0?

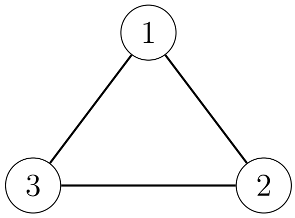
Gourves et al. [GLW17] have already shown that Reachable Assignment and Reachable Object are both NP-hard on general graphs. They have shown further that Reachable Assignment is decidable in polynomial time if G is a tree. Huang and Xiao [HX19] showed that if the underlying graph is a path, then Reachable Object can be solved in polynomial time. Moreover, they studied a version of Reachable Object that allows weak preference lists, i.e. an agent can be indifferent between different objects. Contributing to the Reachable Object problem, Saffidine and Wilczynski [SW18] propose an alternative version of Reachable Object, which is called Guaranteed Level Of Satisfaction, where an agent is guaranteed to obtain an object at least as good as a target object, which is given as an input, in a sequence of swaps $(\sigma_0, ..., \sigma_t)$ where in σ_t one cannot perform a rational trade. They show that Guaranteed Level Of Satisfaction is co-NP-hard.

In our work, we will mainly focus on the Reachable Assignment problem for the two special cases where G is either a cycle or a clique. A cycle is a connected graph where each vertex has exactly two neighbors and a clique is a graph where each vertex shares an edge with each other vertex in the graph. We will show that there exists a polynomial-time algorithm to solve Reachable Assignment on cycles and further prove NP-hardness for Reachable Assignment on cliques. For an example for both graph-classes see Figure 1.1 where we display an example for a clique of size 3 which is also a cycle of the same size and the preference profile of the displayed instance.

At the core of the hardness proof for cliques we will propose a reduction from Reachable Object on cliques for which NP-hardness has been shown by Bentert et al. [Ben+19].

1.2 Introduction to Reachable Assignment on Cycles

In the following chapters we will propose a polynomial-time algorithm for Reachable Assignment on cycles. Recall that the input of Reachable Assignment on cycles



1: $\blacktriangle \succ \boxed{\bullet}$
2: $\blacksquare \succ \boxed{\blacktriangle}$
3: $\bullet \succ \blacktriangle \succ \boxed{\blacksquare}$

Figure 1.1: Example for Reachable Assignment on a cycle with $n = 3$ and preference lists on the right hand side. For the sake of simplicity, the target assignment σ is simply the most preferred object of each agent. The object inside the box is the object that the corresponding agent initially holds. An exemplary sequence of swaps such that every agent reaches its most preferred object is the following. First, agent 2 and 3 swap their currently held objects. Agent 3 then owns object $\blacksquare$, which it prefers the most and agent 2 owns object $\blacktriangle$. Afterwards, agent 1 can trade object $\bullet$ to agent 2 and receive object $\blacktriangle$ in return. Every agent then owns its most preferred object.

is an instance $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$. We will observe that once an object is swapped into either clockwise or counter-clockwise direction, it is impossible to swap it back into the opposite direction. We will show that if we assign such a direction to every object, then we can verify in polynomial time whether there is a swap sequence such that we can reach assignment σ given that we only swap objects in the given directions. If such a swap sequence exists we will say that the assignment of directions "yields σ". Further we will formally define a set of properties that exactly describe the types of assignments of directions which yield σ. We call these types of assignments of directions *valid*. The actual algorithm that solves Reachable Assignment on cycles in polynomial time will create a 2-SAT formula ϕ such that every solution of that formula corresponds to a valid assignment of directions and vice versa. We separate the steps we have just described into five different chapters. In Chapter 2 we will present our preliminaries and propose a preprocessing step that simplifies preference lists. Chapter 3 is dedicated to introducing the aforementioned assignment of directions formally. Based on this definition we will introduce multiple related concepts that we will need throughout the other chapters.

In Chapter 4 we will propose an algorithm we call *Greedy Swaps* that verifies whether an assignment of directions yields the target assignment σ. The conclusion that should be drawn from that chapter is that an algorithm solving Reachable Assignment on cycles must only be able to decide whether or not there is an assignment of directions that yields σ.

In Chapter 5 we define the set of properties that describes valid direction assignments. It will conclude with the proof that our definition of validity exactly defines the set of assignments of directions that yield σ.

Chapter 6 shows how to construct a 2-SAT formula for which a truth assignment can be mapped to a valid assignment of directions and show the actual algorithm that decides Reachable Assignment on cycles in polynomial time.

Finally, in Chapter 7 we show NP-hardness of Reachable Assignment on cliques and end with a conclusion in Chapter 8.

2 Preliminaries and Preprocessing

In this chapter we will introduce basic formalism in order to speak unambiguously about the REACHABLE ASSIGNMENT problem on cycles. First we define general concepts from graph theory and fix our mathematical notation. We introduce two crucial preparation steps we need to perform for every problem instance. Afterwards we extend the existing notation to concepts widely used in our work.

We will use graph-theoretical concepts similar to Diestel [Die12]. Further let $G := (V, E)$ be a graph. Let $W \subseteq V$ be a subset of V. An induced sub-graph $G[W]$ is a sub-graph $G' := (W, E')$ of G where for every edge $(p, q) \in E$, it holds that $(p, q) \in E'$ if and only if $p \in W$ and $q \in W$.

We will further use the notation $[a, b]$ to denote the set of integers $\{a, a + 1, ..., b\}$.

Example 1. The integer interval $[3, 6]$ is equal to the set $\{3, 4, 5, 6\}$.

2.1 Cyclic Sequences: Formalizing Directions on a Cycle

We are now going to formalize the directions *clockwise* and *counter-clockwise* in a cycle. Note that in every cycle there are exactly two complementary paths from one agent to another. First we define the following function.

Definition 2.1. Let $n \in \mathbb{N}$ be a number. We define the function $h_n : [0, n + 1] \to [1, n]$ as follows:

$$h_n(x) := \begin{cases} 1 & \textit{if } x = n + 1, \\ n & \textit{if } x = 0, \\ x & \textit{otherwise.} \end{cases}$$

Further we define the following sequence based on the number of agents n.

Definition 2.2. Given a number n and numbers $0 < a, b \leqslant n$, then $\mu_{a,b}$ denotes a sequence such that

$$\mu_{a,b} = \begin{cases} (a, a + 1, ..., n, 1, 2, ..., b) & \text{if } b < a \\ (a, a + 1, ..., b) & \text{otherwise} \end{cases} \tag{2.1}$$

Example 2. Let $n = 8$. Then $\mu_{2,7} = (2, 3, 4, 5, 6, 7)$ and $\mu_{7,2} = (7, 8, 1, 2)$.

In the following, we assign each agent a number from the integer interval $[1, n]$ such that for every agent named $i \in [1, n]$ its clockwise neighbour is named $h_n(i + 1)$ and its counter-clockwise neighbor is called $h_n(i - 1)$. Now we can use cyclic sequences to

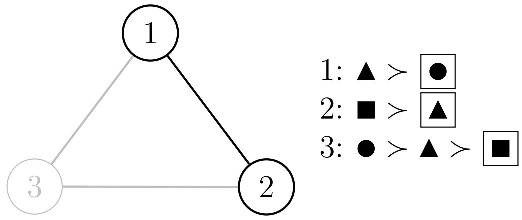


Figure 2.1: Example for Reachable Assignment on a cycle with $n = 3$ and cyclic sequence $\mu_{1,2} := (1, 2)$. The sub-path drawn in black is the induced sub-path of C_3 with respect to $\mu_{1,2}$.

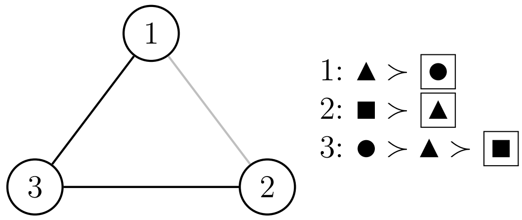


Figure 2.2: Example for Reachable Assignment on a cycle with $n = 3$ and cyclic sequence $\mu_{2,1} := (2, 3, 1)$. The sub-path drawn in black is the induced sub-path of C_3 with respect to $\mu_{2,1}$.

describe paths of C_n. Henceforth, if we write the sequence $s := \mu_{i,j}$, then we will refer to a set of agents. Note that then $C_n[s]$, the induced sub-path of C_n by the agents in the sequence s, is the path in C_n from agent i to agent j in clockwise direction. For an example of how we can apply cyclic sequences to cycles see Figures 2.1 and 2.2.

2.2 Reducing Preference Lists

An important aspect of Reachable Assignment in general is that agents only agree on swaps if they prefer the other object over the object they are currently holding. Whenever an agent prefers an object o over the object p it is assigned in σ, then we can simply remove o from its preference lists. This is due to the fact that once the agent possesses o, then it cannot receive p anymore as it prefers o over p and it will, by definition of rational trades, only receive more preferred objects in the future.

Reduction Rule 2.2.1. *For every agent i with object $p := \sigma(i)$ do the following. If there exists an object q such that*

$$q \succ_i p \tag{2.2}$$

then remove q from $\succ_i$.

We will assume that every instance has already been preprocessed by Reduction Rule 2.2.1.

2.3 Initial Position of Objects

In this section we will introduce an important formalism we will use throughout this work. Recall that assignments are bijections. This allows us for any assignment σ', any object y and the agent i with $\sigma'(i) = y$ to denote i by $\sigma^-1(y)$.

Note that with σ we denote the target assignment whereas σ_0 denotes the initial assignment of the instance. We will also sometimes speak about some assignment after a sequence of i swaps as σ_i. If performing the i swaps of a swap sequence S, starting in assignment σ_j, results in an assignment σ_{j+i} we say that S *transforms* σ_j into σ_{j+i}.

Using the concept of inverse assignments we can define a notation that allows use to describe the set of objects that are initially held by the agents of some sub-path of cycle C_n.

Definition 2.3. Let q and t be two objects. Let $i := \sigma_0^{-1}(q)$ and let $j := \sigma_0^{-1}(t)$. Then we define the set $\Delta_{q,t}$ as

$$\Delta_{q,t} := \{x \in X \mid \sigma_0^{-1}(x) \in \mu_{i,j}\} \tag{2.3}$$

and call $\Delta_{q,t}$ the *object domain of* $\mu_{i,j}$.

Note that by construction it holds that

$$\forall i, j \in N . \mu_{i,j} \cup \mu_{j,i} = V(C_n) = N \tag{2.4}$$

and thus also

$$\forall q, t \in X . \Delta_{q,t} \cup \Delta_{t,q} = X. \tag{2.5}$$

3 Directions in a Cycle

In this chapter we will formally introduce the aforementioned assignments of directions. We will refer to them as *selections* and denote them always with γ. In Section 3.1 we will first motivate why it is resourceful to assign a specific direction to an object which in particular means that we will not allow an object to, for example, be swapped into clockwise and then into counter-clockwise direction. Afterwards we show in Section 3.2 that the number of times that two objects can be swapped is bounded by two. Even more general, we will show that if two objects are assigned opposite directions then there are at most two edges where the corresponding preference lists of the incident agents allow a rational trade of objects.

3.1 Direction Assignments

Consider an instance $\mathcal{I} := (N, X, \succ, G, \sigma_0, \sigma)$ and the target assignment σ. Let p be an object, let j be the agent such that $\sigma(j) = p$ and let i be the agent such that $\sigma_0(i) = p$. Since the underlying graph is a cycle, there are exactly two paths for p to get to j, its destination. By definition, once p has been swapped, say from agent i to agent $i+1$ then p is not able to return to agent i since agent i just received an object that it prefers over p and will therefore not accept p again. Hence, if p is swapped again, then it is given to agent $i+2$ and the argument can be repeated for agent $i+1$. As there are only two paths between agents i and j, there are also only two directions, namely clockwise and counter-clockwise. We will henceforth code these directions into a binary number such that we say that the direction of p is equal to 1 if p is swapped in clockwise direction and equal to 0 if p is swapped in counter-clockwise direction. We will now generalize this for every object as follows:

Definition 3.1. Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be an instance of Reachable Assignment on cycle C_n. Let γ be a function that assigns each object in X a number in $\{0, 1\}$. We call such a function a *selection* of $\mathcal{I}$.

Let further p be an object. Then $\gamma(p)$ denotes the direction that p is assigned by γ.

Given a selection we will now formalize the distance of two objects or an object and an edge relative to a given selection.

Definition 3.2. Let γ be a selection and let q be an object. Let q' be another object and let e be an edge. The *distance* of q and q' is defined as follows.

$$\operatorname{dist}_\gamma(q, q') := \begin{cases} |\Delta(q, q')| & \gamma(q) = 1 \\ |\Delta(q', q)| & \textit{otherwise} \end{cases} \tag{3.1}$$

We say that q' is closer to q than q'' with respect to selection γ if $\text{dist}_\gamma(q, q') < \text{dist}_\gamma(q, q')$. Let e be incident to agents i and j. Let $\sigma_0(i)$ be closer to q than $\sigma_0(j)$. The *distance* of q and e is defined as follows.

$$\text{dist}_\gamma(q, e) := \begin{cases} |\Delta(q, \sigma_0(i))| & \gamma(q) = 1 \\ |\Delta(\sigma_0(i), q)| & \textit{otherwise} \end{cases} \tag{3.2}$$

We say that edge e is closer to q than edge f with respect to selection γ if $\text{dist}_\gamma(q, e) < \text{dist}_\gamma(q, f)$.

Example 3. Let $n = 8$ and let $\mathcal{I} := (N, X, \succ, C_8, \sigma_0, \sigma)$ be an instance of Reachable Assignment on cycle C_8. Let further R be an arrangement of two objects x_2 and x_4 that assigns x_2 to agent 3 and x_4 to agent 6. Let γ be a selection that assigns x_2 to clockwise direction and let γ' be a selection that assigns x_2 to counter-clockwise direction. To compute distances $\text{dist}_\gamma(x_2, x_4)$ and $\text{dist}_{\gamma'}(x_2, x_4)$ we first need to compute the corresponding object domains. The cyclic sequence of 3 and 6 in clockwise direction is $(3, 4, 5, 6)$. The cyclic sequence of 3 and 6 in counter-clockwise direction is $(3, 2, 1, 8, 7, 6)$. Since we are not interested in the actual object domains and just in their size, it is sufficient to compute the size of the cyclic sequences, which is equal to the size of the corresponding object domain. Therefore in the case where x_2 is assigned clockwise direction the distance of x_2 and x_4 is equal to 4 and in the case where x_2 is assigned counter-clockwise direction the distance of x_2 to x_4 is equal to 6.

3.2 Properties of Objects in a Direction-Assignment

Now that we have formally defined selections we are interested in using that definition to formally speak about the position of a set of objects, the paths of an object to its destination with respect to the assigned direction in a selection γ and the intersection of these paths, given two objects.

3.2.1 Object Positions Relative to a Direction-Assignment

The following definition will enable us to speak more generally of the position of a set of objects on a cycle.

Definition 3.3. Let $O := \{o_1, ..., o_m\} \subseteq X$ be a set of objects and let $A \subseteq N$ be a set of agents. Let further hold that $|O| = |A|$. We call every bijective relation $R \in O \times A$ an *arrangement of the objects in* O or an *arrangement of the objects* $o_1, ..., o_m$ if for every i it holds that, o_i is on the preference list of $R(o_i)$.

If further

$$\forall i. R(o_i) = \sigma_0^{-1}(o_i),$$

then we say that R is the *initial arrangement of* O.

Example 4. Recall the setup in Figure 1.1. In this scenario, $R := \{(1, \bullet), (2, \blacksquare)\}$ is an arrangement of $\bullet$ and $\blacksquare$ since $\bullet$ appears on the preference list of agent 1 and $\blacksquare$ appears on the preference list of agent 2. Further $R' := \{(1, \bullet), (2, \blacktriangle)\}$ is an initial arrangement since the initially assigned object of agent 1 is $\bullet$ and the initially assigned object of agent 2 is $\blacktriangle$. Lastly $R'' := \{(1, \blacktriangle), (2, \bullet)\}$ is not an arrangement at all since $\bullet$ does not appear on the preference list of agent 2 which means that $\bullet$ can never be assigned to agent 2 in a rational trade.

An arrangement is a more general version of an assignment. We could, for example, express σ_0 as the initial arrangement of the set X, which is the set of objects in a given instance of Reachable Assignment on cycles. Note that we do not allow an assignment where an agent holds an object it could never receive, since it is not on that agent's preference list. An arrangement is useful when we are only interested in partial information about an assignment σ_i, especially information about the position of just a subset of objects. The next definition is useful when we are given arrangements R_0 and R_1 of two distinct sets of objects O_0 and O_1 and want to express a property for every arrangement where the objects in O_0 are positioned according to R_0 and the objects in O_1 are positioned according to R_1.

Definition 3.4. Let R_0 and R_1 be the arrangements of two sets of objects O_0 and O_1. Let further A_i denote the set of the agents of R_i for $i \in \{0, 1\}$. We call $R_0 \cup R_1$ a *joint arrangement* of O_0 and O_1 with respect to R_0 and R_1 if and only if

$$(O_0 \cap O_1) \cup (A_0 \cap A_1) = \varnothing. \tag{3.3}$$

Note that if equation 3.3 is not satisfied, then $R_0 \cup R_1$ is not an arrangement according to the definition above. The equation simply expresses that the union of both sets of objects as well the union of both sets of agents must be distinct which ensures that $R_0 \cup R_1$ remains a bijection.

3.2.2 Remaining Paths of Objects

In this subsection we will start to talk about paths of objects. Given some selection we have already seen that each object is assigned one of two complementary paths on the cycle. Further we introduced selections which assign one of those two paths to each object. We are now interested in describing the sub-path P that an object, that is currently assigned to some agent i, has to pass in order to get to its destination. The agent i does not necessarily have to be the agent that the object was assigned to in the initial assignment σ_0. This can be formalized as follows.

Definition 3.5. Let γ be a selection for instance $\mathcal{I}$. Let R be a arrangement of an object o. Let further $i := R(o)$ and let j be the agent which is the destination of o, i.e. $\sigma^{-1}(o)$. Let $I := \mu_{i,j}$ and $J := \mu_{j,i}$. Then we define the *path of o in arrangement R* as

$$\xi_\gamma^R(o) := \begin{cases} C_n[I] & \text{if } \gamma(o) = 1 \\ C_n[J] & \text{otherwise} \end{cases} \tag{3.4}$$

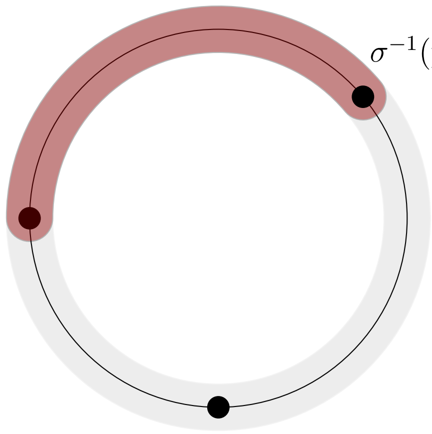


Figure 3.1: An object p with its path marked in red. It has already been moved from its initial position $\sigma_0^{-1}(p)$. Its current position can be described with an arrangement R.

If R is the initial arrangement of o we can also write $P_\gamma(o)$, i.e. the path an object attempts to get to its destination if it was assigned direction $\gamma(o)$. We will then call $P_\gamma(o)$ the *path* of o given selection γ.

Let p be another object such that $\gamma(o) \neq \gamma(p)$. Let R' be a joint arrangement of o and p with respect to arrangements R and S. and let $i := R(o)$ and $j := S(p)$. We define the *shared path of o and p in arrangement R'* as

$$\xi_\gamma^{R'}(o, p) := \xi_\gamma^R(o) \cap \xi_\gamma^S(p) \tag{3.5}$$

Examples for paths and shared paths can be found in Figures 3.1 and 3.2.

3.2.3 Deriving the Bounded Number of Swaps between Objects

In this sub-section we will show that if two objects o and p are assigned opposite directions by some selection γ, then there either exist at most two edges where o and p can be swapped according to the corresponding preference lists of the agents that perform the swap or γ does not yield target assignment σ. This is the main-result of this chapter.

For the proofs in the rest of this chapter and in subsequent chapters we are going to need the following notion of the path that two objects move towards on until they meet at an edge or reach their destinations.

Definition 3.6. Let γ be a selection for instance $\mathcal{I}$ and let p and q be two objects with $\gamma(p) \neq \gamma(q)$. Let R be a arrangement of p and q. Let P be the path from $R(p)$ to $R(q)$ in direction $\gamma(p)$. Then we call P the *swap space* between p and q in arrangement R and denote it as $\omega_\gamma^R(p, q)$. We call $\omega_\gamma^R(p, q)$ *connected* if $\xi_\gamma^R(p, q) \neq \varnothing$ and *disconnected* else.

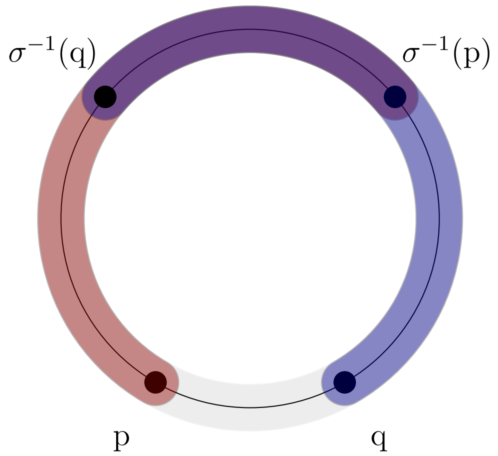


Figure 3.2: Example for two objects p and q. Object p's path is marked red whereas q's path is marked blue. The violet area at which these paths intersect is their *shared path*.

To support the understanding of swap spaces visually, see Figures 3.3 and 3.4.

Before we come to the main-result of this chapter we prove the following intermediate lemma.

Lemma 3.7. *Let γ be a selection for instance $\mathcal{I}$ and let p and q be two objects with $\gamma(p) \neq \gamma(q)$. If there exist a arrangement R of p and q where $\xi^R_\gamma(p, q)$ is non-empty and there exist not exactly one edge on $\xi^R_\gamma(p, q)$ such that the corresponding preference lists allow p and q to be swapped there then γ does not yield target assignment σ.*

Proof. Let us first note that the corresponding swap space of p and q is connected. We will first show that if their swap space is connected, then they must be swapped at some edge on their shared path, which is non-empty, as otherwise γ does not yield the target assignment σ. Consider an edge on the swap space where p and q meet. This could be any edge that is also on the intersection of their paths with respect to arrangement R. We know that they will eventually meet at some edge since this intersection is part of their path. Let the number of swaps that were already performed be t. Let for one of them, say p, hold $\sigma_t^{-1}(p) = \sigma^{-1}(p)$, i.e p has reached its destination. Then since we applied reduction rule 2.2.1 we know that $p \succ_{\sigma^{-1}(p)} q$. Further consider the case where p and q have both not reached their destination but there is no edge on the shared path where the corresponding preference lists allow a swap of p and q. In both cases p and q can not be swapped. But then at least q can not continue its path to its destination in this direction. Since q can also not change its direction we know that $\sigma^{-1}(q)$ can never reach q and thus σ is not reachable with this selection.

Now that we showed that there has to be at least one edge on the shared path of p and q where the objects can be swapped, we continue with the case where there is more than one edge on their shared path where p and q can be swapped according to the corresponding preference lists. Suppose p and q will meet at some point at one of these

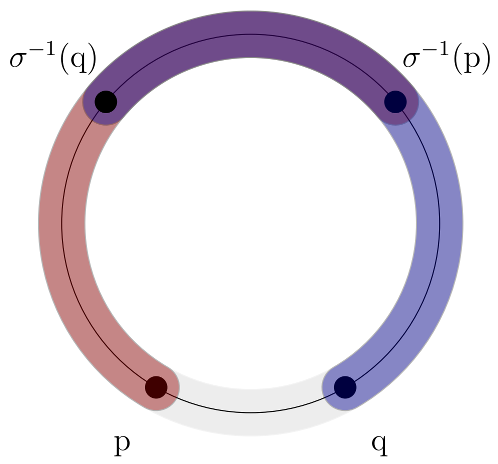


Figure 3.3: Two objects p with $\gamma(p) = 0$ and q with $\gamma(q) = 1$ and their connected swap space.

edges, let us call it e and one of the other edges f. Then since both e and f are on their shared path at least one of p or q must have already passed f. Now consider what it means that objects p and q can be swapped at edge f according to the corresponding preference lists. It means that at edge f, consisting of the two agents k and l it holds:

$$(p \succ_k q) \wedge (q \succ_l p)$$

Let us assume that p has already passed edge f. Then it has passed both k and l. Now since f is on the shared path of p and q, by definition, it is also on the path of q and thus q needs to pass agents k and l in order to get to its destination, since it also cannot change its direction as we have seen. But since $p \succ_k q$, k will not accept q anymore as it has already held p and thus q cannot reach its destination and γ does not yield target assignment σ. The case where q has already passed edge f is of course symmetrical. From this it follows that if there is less or more than one edge on the shared path of p and q then σ is not reachable with γ. From this our statement follows. □

What we have just seen is that given a selection γ and an arrangement, if the corresponding swap space of two objects is connected, i.e. they will have to pass each other eventually to get to their destinations, then there exists exactly one edge on their shared path where the objects can be swapped or γ does not yield target assignment σ. We will therefore define this unique edge formally such that we can refer to it later.

Definition 3.8. Let γ be a selection for instance $\mathcal{I}$ and let p and q be two objects with $\gamma(p) \neq \gamma(q)$. Let there further be a arrangement R of p and q such that $\xi_\gamma^R(p,q)$ is non-empty. Then $e_\gamma^R(p,q)$ denotes the unique edge on $\xi_\gamma^R(p,q)$ where p and q can be swapped. If such an edge does not exist or there are multiple such edges then $e_\gamma^R(p,q)$ is undefined as then σ is not reachable with γ.

Moving on, we prove that given a selection γ there exist at most two edges where two objects can be swapped at or otherwise γ does not yield the target assignment σ.

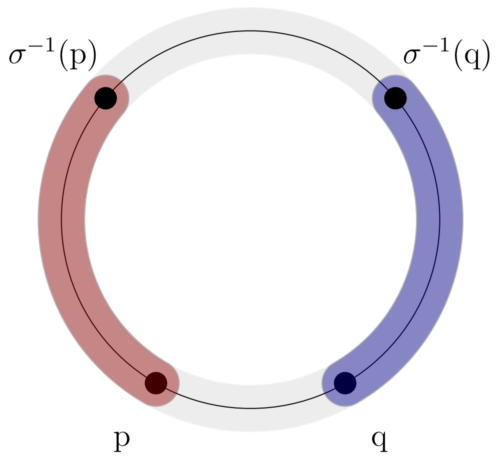


Figure 3.4: Two objects p with $\gamma(p) = 0$ and q with $\gamma(q) = 1$ and their disconnected swap space.

Proposition 3.9. *Let p and q be two objects. For every selection γ there exist at most two edges where p and q can be swapped at or γ does not yield target assignment σ.*

Proof. In this proof, we will just cover the selection where p and q are assigned opposite directions as otherwise the objects cannot be swapped at all. Let R be the initial arrangement of p and let S be the initial arrangement of q and let T be the joint arrangement of p and q with respect to R and S.

We will only look at the case where $\omega_\gamma^T(p, q)$ is connected since otherwise p and q will reach their destinations before being able to meet at an edge (since their paths do not intersect) and therefore will not swap for this γ. Using lemma 3.7 we know that if $\omega_\gamma^T(p, q)$ is connected then there is exactly one edge e, which lies on their shared path, where p and q can be swapped at. Assume that p and q are actually swapped at e yielding the new assignment σ_i. Then we need to compute a new swap space in arrangement T' where $T'(p) = \sigma_i^{-1}(p)$ and $T'(q) = \sigma_i^{-1}(q)$.

Our argument will now go as follows. For p and q to perform the first swap along edge e both of them must have at least passed one agent each. Now we will show that if $\omega_\gamma^{T'}(p, q)$ is connected then $V(C_n) = V(\omega_\gamma^{T'}(p, q))$. If that is true then for p and q to meet again on swap space $\omega_\gamma^{T'}(p, q)$ they must pass n agents combined. If we repeat this for a third arrangement T'' then we know again that $V(C_n) = V(\omega_\gamma^{T''}(p, q))$ if T'' is connected with the same argument as for T'. If p and q then are swapped again they must have passed $2 + n + n$ agents combined and thus at least one of them has passed more than n agents which would mean that it has passed one agent twice, which we already know is not possible. Thus, such a connected T'' cannot exist and there are at most two edges at which p and q can be swapped. We will now show the above statement: If $\omega_\gamma^{T'}(p, q)$ is connected then $V(C_n) = V(\omega_\gamma^{T'}(p, q))$. Recall that T' is the arrangement we compute immediately after p and q are swapped. The swap space is, by definition, the path from $T'(p)$ to $T'(q)$ in the direction of p. Further $T'(p)$ to $T'(q)$ are adjacent and the edge over which they are adjacent is in the opposite direction of p, since p has just passed that edge. Therefore the path from $T'(p)$ to $T'(q)$ in the direction of p covers every agent on the C_n. From this our statement follows. □

As we did after Lemma 3.7 we will introduce notation to refer to the set of edges where two objects p and q can be swapped at a later point.

Definition 3.10. Let p and q be two objects with initial arrangement R and let γ be a selection. Then $E^R_\gamma(p,q)$ is the set of edges were p and q can be swapped according to the preference lists of the agents incident to these edges. Proposition 3.9 tells us that if σ is reachable then

$$|E^R_\gamma(p,q)| \leqslant 2 \tag{3.6}$$

Further if $\gamma(p) = \gamma(q)$ then

$$E^R_\gamma(p,q) = \varnothing. \tag{3.7}$$

We will also refer to $E^R_\gamma(p,q)$ as $E_\gamma(p,q)$ if R is the initial arrangement of p and q.

Concluding this chapter, we have formally introduced the assignment of directions to objects as selections. Further we introduced arrangements which describe the position of a set of objects. Lastly we showed Proposition 3.9 which states that given a selection γ, two objects can be swapped at most at two edges or otherwise γ does not yield target assignment σ.

At the very beginning of this chapter, when we derived the notion of selections we showed that an object, once swapped into one direction, cannot change its direction due to the nature of preference lists. Therefore one selection represents one out of 2^n distinct assignments of directions to objects. Our task remains to find a selection that actually yields the target assignment σ or to be sure that there exists no such selection. We will address this in the following chapter, where we will propose a polynomial time algorithm that decides whether for a given selection yields σ. We do so by giving an exact characterization of all selections that yield σ. Afterwards we will show in Chapter 6 how to compute such a selection, if it exists.

4 Why Swap Order Does Not Matter

In this chapter, we propose the algorithm *Greedy Swaps* and show that it decides whether an assignment σ can be reached if objects are traded only in the directions assigned to them by some selection γ which is the input of the algorithm. As the name already reveals, *Greedy Swaps* is a greedy algorithm. It simply checks whether there is an object that has not reached its destination and then tries to swap it according to its assigned direction. If the swap fails, the algorithm decides that for the given selection, the target assignment σ cannot be reached.

In Section 4.1 we will define how we can detect two objects for which we know that they must be swapped as otherwise σ is not reachable with the given selection. Afterwards we will take a look at a special type of selections for which we can immediately decide whether σ is reachable or not with these selections. In Section 4.2 we will state the *Greedy Swaps* algorithm and prove its correctness.

4.1 Objects In Swap Position

Given a selection γ we will say that object p *walks* or *moves* into direction $\gamma(p)$. An important observation is that two objects p and q in some assignment σ', which is not necessarily the initial assignment σ_0 or the target assignment σ, can only be swapped if their current agents are neighbors. We observe further that p and q can only be swapped if the directions that were assigned to them by a selection γ, must indicate that p, if it is swapped again, will be received by the agent that is currently holding object q and that q, if it is swapped again, will be received by the agent that is currently holding object p. We will formalize this notion using the following definition.

Definition 4.1. Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be an instance of Reachable Assignment on cycle C_n and let γ be selection of $\mathcal{I}$. Let there further be agents i holding object p and j holding q. Let j be i's neighbour in clockwise direction. If $\gamma(p) = 1 = 1 - \gamma(q)$ then we say that *p and q are facing each other* and if also $\sigma(i) \neq p$ or $\sigma(j) \neq q$, then we say that *p and q are in swap position* and denote that by $\rho_\gamma(p, q)$.

Figure 4.1 shows an example of a C_3 where objects ▲ and ● are in swap position. Note that in Definition 4.1 it suffices that one of the objects has not yet reached its destination. This is because we want to ensure that if no two objects are in swap position in some assignment σ', then σ' is equal to σ, the target assignment. This ensured as, if for assignment σ' and a selection γ there exist no two objects in swap position, then it follows from definition 4.1 that at least one of the two following statements is true:

1. There exists no object that has not reached its destination.

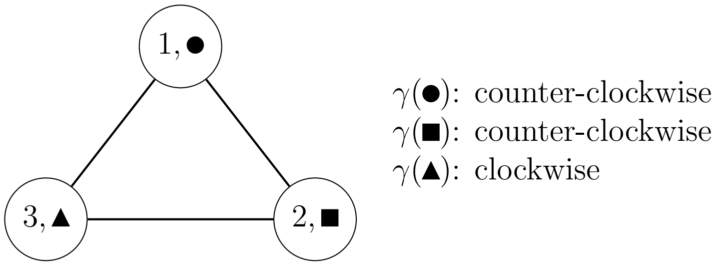


Figure 4.1: Example for two objects in swap position. We assume that none of the three objects has reached its destination. Object ▲ is assigned clockwise direction and will therefore pass agent 1 next on its path. Object ● is assigned counter-clockwise direction and will therefore pass agent 3 next. The objects ▲ and ● are therefore in swap position. Agent 2 is also a neighbor of agent 3. Further the object assigned to agent 2, namely ■, and the object assigned to agent 3, namely ▲, are also assigned opposite direction. However, the two objects are not facing each other and are therefore not in swap position.

2. There exist no two objects that are facing each other.

These statements follow directly from the definition as they are the two statements that must be true for two objects to be in swap position. The first statement can also be equivalently expressed as: "Every object has reached its destination", which of course means that the target assignment σ was reached. So if no two objects are in swap position for the current assignment σ' and not every object has reached its destination, then the second statement must be true. Note that the second statement is equivalent to: "Every object is assigned the same direction". Recall that the directions are assigned by the selection γ. Thus, if every object is assigned the same direction in σ', then only assignment σ_0 itself is reachable from σ_0. If $\sigma_0 \neq \sigma$, then that means that γ does not yield target assignment σ and otherwise every selection γ does yield σ.

4.2 Stating the Algorithm

In this section we will show that Algorithm 1 decides whether a given selection γ yields the target assignment σ. We have already defined selections as an assignment of directions to the set of objects in an instance $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ of Reachable Assignment on cycle C_n. Algorithm 1 takes a selection γ as its input and computes whether σ can be reached with γ. In the following proposition, we will prove that deciding Reachable Assignment on cycles can be reduced to finding a selection such that Algorithm 1 returns *True*.

Proposition 4.2. *Let $\mathcal{I}$ be a* Reachable Assignment *instance on cycle C_n. Algorithm 1 returns True if and only if γ yields σ.*

Proof. Assume that for a given selection γ, Algorithm 1 returns *True* but γ does not yield σ. Then the algorithm has performed a sequence of swaps that yields a situation

Data: Reachable Assignment instance $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ on cycle C_n and selection γ

```
if σ0 = σ then
    return True;
end
if ∃d ∈ {0,1}.∀o ∈ X.γ(o) = d then
    return False;
end
σ' ← σ0;
while ∃x1, x2 ∈ X.ρ_γ(x1, x2) do
    i ← σ'^{-1}(x1);
    j ← σ'^{-1}(x2);
    if x1 ≻_j x2 and x2 ≻_i x1 then
        Swap x1 and x2;
        Update σ';
    else
        return False;
    end
end
return True;
```

Algorithm 1: Greedy Swaps

where, by definition of swap positions, either every object has reached its destination, which is a contradiction since σ was not reached, or for every object p held by agent i and the object q held by the neighbour of i in direction $\gamma(p)$, it holds: $\gamma(p) = \gamma(q)$. Note, however, that this implies that every object in the set of objects X in $\mathcal{I}$ is assigned direction $\gamma(p)$ and therefore the Algorithm 1 returns *False*, a contradiction.

Now suppose that a selection γ yields assignment σ, but Algorithm 1 returns *False*. Then either $\sigma_0 \neq \sigma$ and every object is assigned the same direction, or there are two objects p and q, which are in swap position at some edge e, but the corresponding preference lists do not allow a swap. In the former case γ clearly does not yield σ. In the latter case, consider the initial arrangement R of p and q and the corresponding shared path $\xi^R_\gamma(p, q)$ for the initial assignment σ_0. We know that it is defined since p and q are in swap position at the point where Algorithm 1 returns *False* and thus $\gamma(p) \neq \gamma(q)$ and we know that their shared path is non-empty because p and q meet at some edge during the execution of Algorithm 1 but have not both arrived at their destination.

Let m be the number of objects that are initially assigned to an agent on the swap space of p and q in their initial arrangement R and are assigned direction $\gamma(p)$. We will now show that q has to be swapped with exactly $m - 1$ objects before q and p can be swapped. Starting from the agent that initially holds q we can then calculate the edge e' where q and p meet in every swap sequence that results in reaching the target assignment σ.

Suppose q can be swapped with more than $m-1$ objects before meeting p at edge e'.

Then q must either be swapped with at least one object q' that is not initially assigned to an agent on the swap space of q and p. This, however, means that q will meet q' after p, a contradiction. Otherwise q must be swapped with more objects that are initially assigned to an agent on the swap space of q and p, than there are objects on that swap space with the opposite direction of q, which is not possible as two objects can only be swapped if they are assigned different directions, a contradiction. Now suppose that q can be swapped with less than $m - 1$ objects before meeting p. Then there is an object r starting on the swap space of p and q with $\gamma(r) \neq \gamma(r)$ that does not swap with q. However then q will not be able to meet p as at least r is between them, a contradiction.

Now that we know that q will swap exactly m objects before meeting p, we can also determine the edge f where, given γ, they must necessarily meet, in every swap sequence where objects are swapped according to the directions assigned to them by γ. Now since we assumed that σ is reachable with selection γ, then there must be a swap sequence where q and p must be swapped at edge f. But since p and q met, by assumption, at edge e, either $e = f$, which is a contradiction because at e the preference lists of the incident agents do not allow a swap between p and q or otherwise p and q can not have met at edge e, if all swaps were performed according to the directions assigned by selection γ, contradiction. □

Starting in the subsequent chapters, we will first define exactly what properties a selection must have in order for Algorithm 1 to return *True*. Recall that before stating Algorithm 1 we used the term *valid* to refer to a selection γ for which there exists a swap sequence, where every object is only swapped in the direction assigned to it by γ, which transforms σ_0 to σ. Note that if Algorithm 1 returns *True* on γ, then we are given such a swap sequence. Otherwise we know that such a swap sequence cannot exist for γ. Therefore a selection is valid if and only if Algorithm 1 returns *True* on it. Hence, our next goal is to define validity and show that if a selection is valid, according to that definition, then Algorithm 1 returns *True*. We will attempt this in the next chapter, Chapter 5.

5 Characterizing The Instances That Have a Solution

In the previous chapters we have shown that each object has exactly two possible paths from its initial position to its destination. These paths are determined by the direction of the first swap an object is involved in. If it is moved in clockwise direction, it is impossible afterwards to be involved in a swap that moves it in counter-clockwise direction and vice versa. In the last chapter we have seen that if we are assigning one of the two possible directions to each object, then we can verify in polynomial time whether there exists a swap sequence that transforms σ_0 into σ for this assignment of directions. The verification can be done by Algorithm 1. In this chapter, we will provide a crucial building block for solving Reachable Assignment on cycles. In Section 5.2 we will define a set of properties that a *selection* needs to fulfill in order for Algorithm 1 to return *True* and then show in Section 5.3 that these properties are both necessary and sufficient.

We will use this result to solve Reachable Assignment on cycles in Chapter 6 by constructing selections in polynomial time that suffice the characterization presented in this chapter or decide that it is impossible to find such a selection for a given instance of Reachable Assignment on cycles.

5.1 A Property To Determine That Every Object Meets At Every Edge At Most One Object

In this section we will formalize a sequence of mathematical objects on top of which we will define valid selections in the next section. We begin by defining a set H as follows.

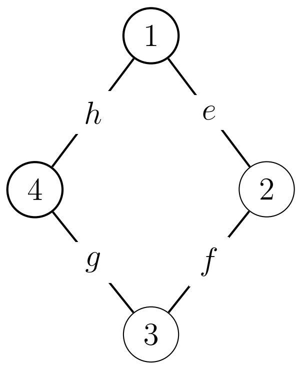
Definition 5.1. Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be an instance of Reachable Assignment. Then we define

$$H := X \times E(C_n).$$

Further we call an element (p, e) of H an *object-edge-pair*.

Recall that each object has two complementary paths from its initially assigned agent to its destination in target assignment σ and that the union of these paths covers the whole cycle C_n. We will use this to formalize the following.

Definition 5.2. Let $(p, e) \in H$ be an object-edge-pair. We define the function $d : H \to \{0, 1\}$ which assigns the direction c to (p, e) such that e lies on the path of p from its



1: $x_3 \succ x_2 \succ \boxed{x_1}$
2: $x_1 \succ x_3 \succ x_4 \succ \boxed{x_2}$
3: $x_4 \succ x_1 \succ \boxed{x_3}$
4: $x_2 \succ x_3 \succ \boxed{x_4}$

Figure 5.1: Example for Reachable Assignment on a C_4 with edges e, f, g, h. The candidate list of x_1 in clockwise direction is $C(x_1, e) := \{x_2, x_3\}$, since the path of x_1 in clockwise direction contains only edge e. The candidate lists of x_1 in counter-clockwise direction are $C(x_1, h) := \emptyset$, $C(x_1, g) := \emptyset$ and $C(x_1, f) := \{x_4\}$. Note that since x_1 does not appear on the preference list of agent 4, both $C(x_1, h)$ and $C(x_1, g)$ are empty.

initially assigned agent to its destination in target assignment σ where p is only swapped in direction c.

Further we define the following.

Definition 5.3. Let $(p, e) \in H$ be an object-edge-pair. We define the set $C(p, e)$ as

$$C(p, e) := \{q \in X \mid p \textit{ and } q \textit{ can be swapped at } e\}$$

and call $C(p, e)$ the candidate list of p at edge e.

Figure 5.1 illustrates candidate lists in an example of a C_4. This computation is necessary, for example, to determine for two objects p and q whether there exists an edge on the intersection of their paths, where p and q can be swapped.

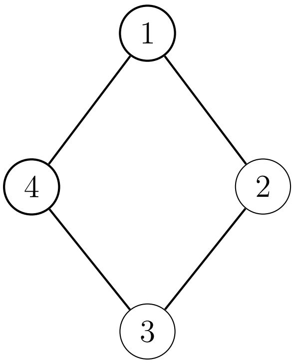
Based on these definitions we will now formalize the following which we will use to define valid selections in the next section.

Definition 5.4. Let γ be a selection and let $(p, e) \in H$ be an object-edge-pair. Then we define the function $f_\gamma(p, e)$ as follows:

$$f_\gamma(p, e) := \begin{cases} |\{q \in C(p, e) \mid \gamma(p) \neq \gamma(q)\}|, & \textit{if } d(p, e) = \gamma(p) \\ 1, & \textit{otherwise.} \end{cases}$$

5.2 A Characterization of Direction Assignments that Yield the Target Assignment

In this section we will define the exact characterization of all selections that yield the target assignment σ. We will refer to every selection that suffices this characterization



1: $x_3 \succ x_2 \succ \boxed{x_1}$
2: $x_1 \succ x_3 \succ \boxed{x_2}$
3: $x_4 \succ \boxed{x_3}$
4: $x_2 \succ x_3 \succ \boxed{x_4}$

Figure 5.2: Example for an instance $\mathcal{I} := (\{1,..,4\}, \{x_1,...,x_4\}, \succ, C_4, \sigma_0, \sigma)$ of Reachable Assignment on cycle C_4. Every selection γ where $\gamma(x_1) = 1$ and $\gamma(x_2) = \gamma(x_3) = 0$ is ambiguous. The edge where the ambiguity occurs is the edge that is incident to agents 1 and 2. Agent 1 prefers both x_2 and x_3 over x_1 and agent 2 prefers object x_1 over both x_2 and x_3. An example for an unambiguous selection is the selection γ' where

$$\gamma(x_1) = \gamma(x_3) = 1 = 1 - \gamma(x_2) = 1 - \gamma(x_4)$$

Further γ' yields target assignment σ. A sequence of swaps that transforms σ_0 into σ is $((x_1, x_2, x_3, x_4), (x_1, x_3, x_2, x_4), (x_3, x_1, x_2, x_4), (x_3, x_1, x_4, x_2))$.

as *valid*. Validity consists of three properties. We call them *unambiguity*, *completeness* and *harmony*. We will define these first individually and then combine them to the definition of validity.

We will start by defining unambiguity and completeness. Afterwards we state a series of definitions that are then combined into the third property of validity, namely harmony.

Definition 5.5. Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be an instance of Reachable Assignment on cycle C_n. We call the selection γ *unambiguous* if and only if

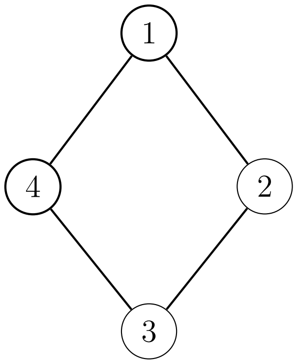
$$\forall (p, e) \in H. f_\gamma(p, e) \leqslant 1.$$

Definition 5.6. Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be a Reachable Assignment instance on cycle C_n. We call the selection γ *complete* if and only if

$$\forall (p, e) \in H. f_\gamma(p, e) \geqslant 1.$$

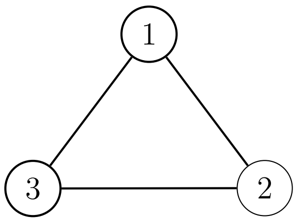
Examples for Definitions 5.5 and 5.6 can be found in Figures 5.2 and 5.3. If a selection γ is both unambiguous and complete, we say that γ is sound. We will continue by defining two properties that are each expressed with respect to two objects to combine these definitions into the definition of harmony afterwards. To define these, however, we need the following definition of opposite objects.

Definition 5.7. Let $d \in \{0, 1\}$ be the selected direction of objects p and q for some instance $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ of Reachable Assignment on cycle C_n. Let γ be



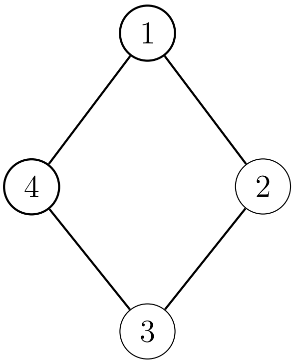
1: $x_3 \succ x_2 \succ \boxed{x_1}$
2: $x_1 \succ x_3 \succ \boxed{x_2}$
3: $x_4 \succ \boxed{x_3}$
4: $x_2 \succ x_3 \succ \boxed{x_4}$

Figure 5.3: Example for an instance $\mathcal{I} := (\{1, .., 4\}, \{x_1, ..., x_4\}, \succ, C_4, \sigma_0, \sigma)$ of Reachable Assignment on cycle C_4. Every selection γ where $\gamma(x_1) = \gamma(x_2) = \gamma(x_3) = 1$ is incomplete. The edge where the incompleteness occurs is the edge that is incident to agents 1 and 2. Agent 1 can swap x_1 with agent 2 in exchange for either x_2 or x_3. However, neither of these objects are assigned the opposite direction. Hence, x_1 cannot be swapped with any object at that edge. An example for a complete selection can be found in Figure 5.2. There, the selection γ' is complete.



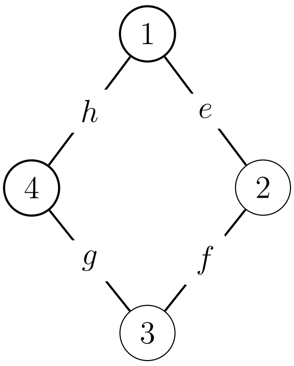
1: $x_3 \succ \boxed{x_1}$
2: $x_1 \succ x_3 \succ \boxed{x_2}$
3: $x_2 \succ x_1 \succ \boxed{x_3}$

Figure 5.4: Example for an instance $\mathcal{I} := (\{1, .., 3\}, \{x_1, ..., x_3\}, \succ, C_3, \sigma_0, \sigma)$ of Reachable Assignment on cycle C_3. For every selection where $\gamma(x_1) = 1 = 1 - \gamma(x_2)$, x_1 and x_2 are incompatible. This is because x_2 is not on the preference list of agent 1. Hence, agent 1 cannot trade x_1 in exchange for x_2 with agent 2, as that trade is not rational.



1: $x_3 \succ x_4 \succ x_2 \succ \boxed{x_1}$
2: $x_1 \succ x_3 \succ \boxed{x_2}$
3: $x_4 \succ \boxed{x_3}$
4: $x_2 \succ x_3 \succ \boxed{x_4}$

Figure 5.5: Example for an instance $\mathcal{I} := (\{1, .., 3\}, \{x_1, ..., x_3\}, \succ, C_3, \sigma_0, \sigma)$ of Reachable Assignment on cycle C_3. Objects x_1 and x_4 are opposite. This is because if x_1 arrives at its destination, agent 2, then x_4 cannot pass x_1 as agent 2 prefers x_1 the most. This is ensured since we applied Reduction Rule 2.2.1 and the agent that prefers an object the most is also its destination, according to target assignment σ.



1: $x_4 \succ x_3 \succ x_2 \succ \boxed{x_1}$
2: $x_1 \succ x_3 \succ \boxed{x_2}$
3: $x_2 \succ \boxed{x_3}$
4: $x_3 \succ \boxed{x_4}$

Figure 5.6: Example for an instance $\mathcal{I} := (\{1, .., 3\}, \{x_1, ..., x_3\}, \succ, C_3, \sigma_0, \sigma)$ of Reachable Assignment on cycle C_3. For every selection where $\gamma(x_1) = \gamma(x_4) = 0$, x_4 shields x_1 in direction 0. It holds that $P_\gamma(x_1) = (h, g, f)$ and $P_\gamma(x_4) = (g, f, e)$. Thus, $P_\gamma(x_1) \cap P_\gamma(x_4) = (g, f)$. However, x_4 is closer to agent 2, the destination of x_1 than x_1 in counter-clockwise direction. Further, agent 4 prefers x_4 over x_1. Thus, x_4 shields x_1 in direction 0.

a selection where $\gamma(p) = \gamma(q) = d$ and let γ' be a selection where $\gamma'(p) = \gamma'(q) = 1 - d$ We say that p and q are opposite if

$$P_\gamma(q) \subset P_\gamma(p)$$

and

$$P_{\gamma'}(p) \subset P_{\gamma'}(q).$$

Figure 5.5 demonstrates an example for Definition 5.7. The next definition is also necessary to define harmonic selections.

Definition 5.8. Let γ be a selection and let p and q be two objects. Let $c \in \{0, 1\}$ be a direction such that it holds that $\gamma(p) = \gamma(q) = c$. If $P_\gamma(p) \cap P_\gamma(q) \neq \emptyset$ and the destination of q is closer to p in direction c than to q it must hold for every agent i on $P_\gamma(p) \cap P_\gamma(q)$:

$$q \succ_i p.$$

Otherwise we say that p shields q in direction c.

An example for definition 5.8 can be found in Figure 5.6. Before we continue, we state the following about opposite objects.

Observation 5.9. *If objects p and q are opposite, then there exists a direction $c \in \{0, 1\}$ such that either p shields q in direction c and q shields p in direction $1 - c$ or vice versa for every selection γ where $\gamma(p) = \gamma(q)$.*

The next definition is the first of two definitions that we need to state in order to define harmonic selections. But first we need the following observation.

Observation 5.10. *Let P, Q be two paths on cycle C_n and let $P \subset Q$. Let further $P^{-1} := C_n \backslash P$ be the complementary path to P on C_n. Then the induced graph $C_n[V(P^{-1}) \cap V(Q)]$ contains two distinct sub-paths P_0 and P_1.*

As an example for this observation, see Figure 5.5. In this example let P be the path $(1, 2)$ and let Q be the path $(4, 1, 2, 3)$. Then $P^{-1} = (1, 4, 3, 2)$, as constructed in Observation 5.10. The intersection of P^{-1} and Q results thus in paths $P_0 = (1, 4)$ and $P_1 = (3, 2)$. The next definition uses this observation to define compatibility of objects as follows.

Definition 5.11. Let p and q be two objects. We distinguish between two cases.

In the first p and q are opposite. Let $c \in \{0, 1\}$ be a direction, such that for every selection γ where $\gamma(p) = 1 - c = 1 - \gamma(q)$, according to Observation 5.10, the intersection of the paths $P_\gamma(p)$ and $P_\gamma(q)$ results in two distinct sub-paths P_0 and P_1. Then we say that p and q are *compatible* in selection γ if and only if for every $i \in \{0, 1\}$ there exists exactly one edge $e \in P_i$ such that $q \in C(p, e)$.

In the second case p and q are not opposite. Then we say that p and q are *compatible* in a selection if and only if there exists exactly one edge $e \in (P_\gamma(p) \cap P_\gamma(q))$ such that $q \in C(p, e)$.

For an example of Definition 5.11 see Figure 5.4. Based on Definitions 5.7, 5.8 and 5.11 we formalize harmonic selections as follows.

Definition 5.12. We call a selection γ *harmonic* if and only if for every object p there is no object q moving in direction $\gamma(p)$ that shields p in that direction, and every object r with direction $1-\gamma(p)$ is compatible with p.

Now that we have defined all of the above properties we can formalize validity for selections as follows:

Definition 5.13. Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be a Reachable Assignment instance on cycle C_n. We call a selection γ of that instance *valid* if and only if it is unambiguous, complete and harmonic.

5.3 Showing that the Definition of Validity is Correct

In this section we will show that our definition of validity is correct, that is, it is both necessary and sufficient for a valid selection γ to yield the target assignment σ. This will be the main-result of this chapter. Before we state and prove the proposition, we show the following intermediate lemma that we will use to prove the correctness of validity afterwards.

Lemma 5.14. *Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be an instance of* Reachable Assignment *on cycle C_n, let γ be a selection, let $(p, e_0), (p, e_1) \in H$ be two object-edge-pairs where $d(p, e_0) = d(p, e_1)$ and let q_0, q_1 be objects with direction $1-\gamma(p)$. Let further $q_0 \in C(p, e_0)$ and let $q_1 \in C(p, e_1)$. Let $h \in \{0, 1\}$ be the index such that q_h starts closer to p in direction $\gamma(p)$ than q_{1-h}. If γ is harmonic, then e_h is closer to p in direction $\gamma(p)$ than e_{1-h}. If e_{1-h} is closer to p in direction $\gamma(p)$ than e_h, then q_h shields q_{1-h}.*

Proof. Considering the preference lists at edges e_{1-h} and e_h we will denote the agents at e_{1-h} by k and $k+1$ and the agents at e_h by m and $m+1$.
Suppose towards a contradiction that e_{1-h} is closer to p than e_h in direction $d(p, e_h)$.

We distinguish between two cases. In the first case γ is harmonic. Then since q_h and q_{1-h} are assigned the same direction, q_h and q_{1-h} are not opposite. Further q_h and q_{1-h} cannot shield each other. Thus, it must hold that

$$q_{1-h} \succ_{k+1} q_h,$$

because $k+1$ is on both $P_\gamma(q_h)$ and $P_\gamma(q_{1-h})$.

In the second case γ is not harmonic and there is some agent that will not accept q_{1-h} since it prefers q_h more and q_{1-h} would be shielded from its destination by q_h. Therefore we will now assume that γ is harmonic and show that then e_h must be closer to p in direction $\gamma(p)$ than e_{1-h}.

Since p can be swapped with q_{1-h} at e_{1-h} we know that

$$p \succ_{k+1} q_{1-h}$$

since q_{1-h} arrives at $k+1$ and is then swapped with p. However e_h is further away from p than e_{1-h} and in order for q_h to be swapped with p at that edge we know

$$q_h \succ_m p$$

We also know that p has already passed agent $k+1$ to get to m and that q_h must at least come to agent $k-1$ since otherwise q_{1-h} could not reach its destination. Since q_{1-h} can be swapped with p at e_{1-h}, its destination is at least agent k. But if q_h must move at least to $k-1$ it also must pass $k+1$ and since p was there first we know that

$$q_h \succ_{k+1} p$$

because otherwise p would shield q_h from its destination which we know cannot be true as γ is harmonic. However, if those terms are true then

$$q_h \succ_{k+1} p \succ_{k+1} q_{1-h} \succ_{k+1} q_h,$$

which cannot be true as $\succ$ is a strict ordering, a contradiction. □

Using these results, we will now prove that, given an instance $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ of Reachable Assignment on cycles, validity is the exact characterization of all selections that yield the target assignment σ.

Proposition 5.15. *Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be an instance of* Reachable Assignment *on cycle C_n and let γ be a selection for that instance. Algorithm 1 returns True on input γ if and only if γ is valid.*

Proof. First we will prove that if γ is valid then Algorithm 1 returns *True*. Let therefore q be an object with path $P := P_\gamma(q)$. We will prove that if γ is valid then q is guaranteed to pass every edge of P. Since q is arbitrarily chosen we can generalize the statement to all objects. Now recall that Algorithm 1 simply performs a swap between two objects that are in swap position over and over again. It only returns *False* if there are two objects in swap position at edge e that cannot be swapped according to the preference lists of the agents that are incident to e. If we prove for every object, at every edge on its path, that the corresponding preference lists always allow a swap, we know that Algorithm 1 returns *True*.

Suppose q has already passed i objects and let R be the arrangement that assigns q to its current agent. Let $Q := \xi_\gamma^R(q)$ be its rest path and let Q' be the path it has already passed, i.e. $Q' := P_\gamma(q) \setminus Q$. Let t_i be the object that q meets at the first edge on Q, e_i, with $\gamma(q) \neq \gamma(t_i)$. Suppose q and t_i cannot be swapped at e_i due to the preference lists of their agents. Then either there exists no object that q can be swapped with at edge e_i in which case γ is incomplete, a contradiction, or there exists object q' with $\gamma(q') \neq \gamma(q)$ that can be swapped with q at e_i but t_i starts closer to q than q'. We will now show that if e_i is not an edge where q and t_i can be swapped, then that leads to a contradiction.

First of all, since γ is harmonic there must be an edge f on the shared path of q and t_i with respect to their initial arrangement where t_i and q can be swapped. We distinguish between two cases. If f is on Q', the path q has already passed, then q has already passed edge f and there exist two objects with direction $1-\gamma(q)$ that q can be swapped with at f. Then, γ would be ambiguous, a contradiction. Otherwise f is on Q but not equal to e_i. But then the edge where t_i and q can be swapped at is further away from q with respect to γ, than the edge where q and q' can be swapped at, even though t_i is closer to q than q', with respect to γ. But then due to Lemma 5.14, γ is not harmonic, a contradiction. Thus we can conclude that q and t_i can be swapped at edge e_i. Since we chose the edge e_i to be arbitrary we can generalize this to all edges on $P_\gamma(q)$. From this follows that q reaches its destination. This can be generalized to all objects and thus, as described above, Algorithm 1 returns *True*.

Now we will show that if γ is not valid, then Algorithm 1 returns *False*. Let γ be incomplete. Then there exists an object p and an edge e on its path where there exists no object q such that p and q can be swapped at e. Hence, if p reaches this edge there is no object that it can be swapped with. Since this edge is on its path the current agent can not be p's destination and hence σ cannot be reached. Algorithm 1 therefore returns *False*.

Let γ be ambiguous. Then there exist objects p, q and r and edge e on their paths such that if p reaches e, then it can be swapped with both q and r. Say p is swapped with q at e. Note that if p and r can be swapped at edge e then r must at least reach both agents incident to e to get to its destination and at one of these two agents, let us call it i it holds

$$p \succ_i r \tag{5.1}$$

since p and r can be swapped there by assumption. But then r cannot be accepted by i anymore since p has already been assigned to both of the agents before r has been assigned to one of them and thus r cannot reach its destination. Hence, σ cannot be reached. Algorithm 1 therefore returns *False*. .

Let γ be inharmonic. We distinguish between two cases. In the first case there exist objects p and q such that q shields p from its destination. That means that p and q walk in the same direction and q stops at its destination i before p can pass i. Since i, however, is on $P_\gamma(p)$, p cannot reach its destination and σ cannot be reached. In the second case there exist object p and q with $\gamma(p) \neq \gamma(q)$ and an arrangement T of p and q such that the corresponding shared path is non-empty but there exists no edge on that shared path where p and q can be swapped. But if their shared path in arrangement T is non-empty, then p and q must meet at some edge e on that path. Since e is on their shared path at least one of p and q cannot have reached its destination when being assigned to an agent incident to e. But since p and q cannot be swapped at e, at least one of p and q can not reach its destination. In both cases Algorithm 1 returns *False*.

If γ is not valid, then it is either incomplete, ambiguous or inharmonic and thus γ yields a *False*-instance of Algorithm 1, a contradiction. $\square$

In this chapter we have defined the exact characterization of selections that yield σ. We called these types of selections *valid*. Afterwards we showed that a selection is valid according to Definition 5.13 if and only if it yields the target assignment σ. In the next chapter we are going to solve Reachable Assignment on cycles by showing how to construct a 2-SAT formula such that every truth assignment of that formula can be mapped to a valid selection in polynomial time.

6 A Polynomial-Time Algorithm For Reachable Assignment On Cycles

In previous chapters we have studied the Reachable Assignment problem on cycles. We determined that in one swap sequence an object can only be either swapped always in clockwise or always in counter-clockwise direction on the cycle. Hence, we defined the notion of *selections*, which are a direction assignment of the set of possible directions to the set of objects X. With Algorithm 1 we stated an algorithm that can check for a given selection in polynomial time whether there exists a swap sequence that transforms the initial assignment σ_0 to the target assignment σ with the constraint that objects are only swapped in the directions that are assigned to them by the selection. Afterwards we defined the exact characterization of selections for which Algorithm 1 returns *True*. If a selections fulfills the set of properties that define this characterization, then we say that the selection is *valid*. Validity mainly consists of the conjunction of three different properties: *unambiguity, completeness and harmony*, each of which addresses a specific constellation of a selection due to which Algorithm 1 will return *False*.

In this chapter we will solve Reachable Assignment on cycles by proving that if a valid selection exists, then it can be found in polynomial time. We will find valid selections by constructing a 2-SAT formula such that every satisfying truth assignment of that formula corresponds to a valid selection. We will further prove that if the constructed 2-SAT formula is unsatisfiable, then there exists no valid selection. This chapter is structured as follows. In section 6.1 we will construct a 2-SAT formula such that every truth assignment of that formula can be mapped in polynomial time to a harmonic selection. In Section 6.2, we will provide a tool to extend the 2-SAT formula to cover *unambiguity* and *completeness*. Thus, in Section 6.3 we will use these results to construct a 2-SAT formula such that every satisfying truth assignment of that formula corresponds to a valid selection and vice versa. Finally, we will show in that section that the algorithm stated runs in polynomial time.

6.1 Constructing a 2-SAT formula of Constraints with At Most Two Objects Involved

In this section we will construct a 2-SAT formula ψ such that a selection is harmonic if and only if it corresponds to a satisfying truth assignment of ψ. This means that we use objects interchangeably with logical variables. Further, the truth value of an object in a satisfying truth assignment of ψ equals the direction it is assigned in the corresponding selection that we can derive from this truth assignment.

This will be the first of six 2-SAT formulas that we will construct whose conjunction gives us a 2-SAT formula ϕ such that a selection is valid if and only if it is a satisfying truth assignment of ϕ. As of the result of Chapter 5, we know that such a formula decides Reachable Assignment on cycles. We construct the formula ψ as follows.

Construction 6.1. Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be an instance of Reachable Assignment on cycle C_n. We construct the 2-SAT formula ψ as follows.

For every pair p, q of objects for which there exists a direction c, such that q shields p in direction c for every selection γ where $\gamma(p) = c = \gamma(q)$, we add the following term to ψ:

$$\begin{cases} p \to \neg q & \text{if } c = 1 \\ \neg p \to q & \text{otherwise.} \end{cases} \tag{6.1}$$

Further, for every pair p, q of objects for which there exists a direction $c \in \{0, 1\}$ such that p and q are not compatible for every selection γ where $\gamma(p) = c = 1 - \gamma(q)$, then we add the following term to ψ:

$$\begin{cases} p \to q & \text{if } c = 1 \\ \neg p \to \neg q & \text{otherwise.} \end{cases} \tag{6.2}$$

We will now prove that ψ is well-defined.

Lemma 6.2. *Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be an instance of* Reachable Assignment *on cycle C_n. Let γ be a selection. Then γ is harmonic if and only if it is a satisfying truth assignment of ψ.*

Proof. We will prove both directions of the statement by contradiction. For the first direction suppose that γ is a harmonic selection but it does not correspond to a satisfying truth assignment of ψ. Then there must exist a clause c in ψ such that c is not fulfilled by the truth assignment corresponding to γ. We distinguish between two cases.

In the first case c is a clause as defined in Equation 6.1. But then there exist two objects p and q and a direction $c_p \in \{0, 1\}$ such that q shields p in direction c_p for every selection γ' where $\gamma'(p) = c_p = \gamma'(q)$ but it holds that $\gamma(p) = c_p = \gamma(q)$. But then, by definition, γ is not harmonic, a contradiction.

In the second case c is a clause as defined in Equation 6.2. But then there exist two objects p and q and a direction $c_p \in \{0, 1\}$ such that p and q are not compatible for every selection γ where $\gamma(p) = c_p = 1 - \gamma(q)$ but it holds that $\gamma(p) = c_p = 1 - \gamma(q)$. But then, by definition, γ is not harmonic, a contradiction.

We will now show the other direction of the statement. Suppose that γ corresponds to a satisfying truth assignment of ψ but it is not harmonic. We again distinguish between two cases.

In the first case there exist two objects p and q and a direction $c_p \in \{0, 1\}$ such that q shields p in direction c_p for every selection γ' where $\gamma'(p) = c_p = \gamma'(q)$. But since γ corresponds to a satisfying truth assignment of ψ, due to the clause defined in Equation 6.1, if $\gamma(p) = c_p$, then $\gamma(q) = 1 - c_p$ and thus, p does not shield q, a contradiction.

In the second case there exist two objects p and q and a direction $c_p \in \{0,1\}$ such that p and q are not compatible in direction c_p for every selection γ' where $\gamma'(p) = c_p = 1-\gamma'(q)$. But since γ corresponds to a satisfying truth assignment of ψ, due to the clause defined in Equation 6.2, if $\gamma(p) = c_p$, then $\gamma(q) = c_p$ and thus, p and q are compatible, a contradiction. $\square$

6.2 Deriving a Tool to Simulate Where Two Objects Meet Given a Direction Assignment

In this section we will derive a mathematical tool to formulate constraints, that ensure unambiguity and completeness for a selection involving more than two objects in a 2-SAT formula. In this section we will first propose a step to subdivide every Reachable Assignment instance $\mathcal{I} := (N, X, \succ, C_n, \sigma_0)$ on cycle C_n into n subproblems and show that the target assignment σ is reachable if and only if at least one of those subproblems has a solution. Afterwards, we will show for a given subproblem how we can reliably compute for every pair of objects p and q the edge e_1 where p and q meet for the first time if p is assigned clockwise direction and q is assigned counter-clockwise direction and the edge e_2 where p and q meet for the first time if q is assigned clockwise direction and p is assigned counter-clockwise direction or decide that either e_1 or e_2 does not exist. Based on those results we will construct a 2-SAT formula such that every satisfying truth assignment of that formula corresponds to a valid selection and vice versa in Section 6.3.

6.2.1 Subdividing the Reachable Assignment Instance into Subproblems

Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be an instance of Reachable Assignment on cycle C_n. If the target assignment σ and the initial assignment σ_0 are not equal, then we have to perform at least one swap to reach the target assignment. Note that since a swap happens between two agents, there exist exactly n edges where the first swap can occur. Thus, we subdivide instance $\mathcal{I}$ into n subinstances $\mathcal{I}_1, ..., \mathcal{I}_n$. A subinstance consists of the instance $\mathcal{I}$ and an edge e_i. We formalize the subproblem as follows.

First Swap Reachable Assignment
Input: An instance $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ of Reachable Assignment and an edge $e \in E(C_n)$
Question: Is σ reachable if the first swap is performed over edge e?

Even though we cannot predict which subinstance yields a solution, we construct the instances $\mathcal{I}_1, ..., \mathcal{I}_n$ such that all possible edges are covered in one of the subinstances. Thus, if we find a solution for one of these subinstances, then we have also found a solution to the original instance $\mathcal{I}$. Suppose that for a subinstance $\mathcal{I}_i$, the first swap happens at edge e with incident agents i and $j = h_n(i+1)$. The object $\sigma_0(i)$ is swapped into clockwise direction and the object $\sigma_0(j)$ is swapped into counter-clockwise direction.

Thus, in every selection γ that yields the target assignment σ as a solution to the subinstance $\mathcal{I}_i$, it must hold that

$$\gamma(\sigma_0(i)) = 1$$

and

$$\gamma(\sigma_0(j)) = 0.$$

For an instance of First Swap Reachable Assignment with edge $(i, h_n(i+1))$ we refer to the pair of objects $x := \sigma_0(i)$ and $y := \sigma_0(j)$ as the *guess* of that instance and denote it $\Phi := (x, y)$, where x is referring to the object that is swapped into clockwise direction and y is referring to the object that is swapped into counter-clockwise direction. For every selection γ where $\gamma(x) = 1 = 1 - \gamma(y)$, we say that γ *respects* the instance of First Swap Reachable Assignment with guess $\Phi := (x, y)$. Further, Φ_0 denotes the object in guess Φ which is swapped in counter-clockwise position and Φ_1 denotes the object in guess Φ which is swapped in clockwise position. We also denote the unique path of x in clockwise direction from its initial agent to its destination with P_x and the unique path of y in counter-clockwise direction from its initial agent to its destination with P_y. If we can prove that for a guess Φ and the corresponding instance of First Swap Reachable Assignment there exists no valid selection that respects Φ, then we say that the guess Φ is *wrong*.

Before concluding this section, we give some definition that we will use throughout the rest of this work. Intuitively, these capture properties of objects which cannot be swapped in one direction given a certain guess Φ.

Definition 6.3. Let $\mathcal{I} := ((N, X, \succ, C_n, \sigma_0, \sigma), e)$ be an instance of First Swap Reachable Assignment on cycle C_n with guess $\Phi := (x, y)$. Let p be an object such that there exists at most one $d \in \{0, 1\}$ such that a selection γ that respects $\mathcal{I}$ and where it holds that

$$\gamma(p) = d$$

can be valid. Then we say that p is *decided* in direction d. Otherwise we say that p is undecided.

Based on Definition 6.3 we will derive an important property for the candidate lists of the guessed objects. The following Lemmas will be useful when we construct a 2-SAT formula such that every satisfying truth assignment of that formula corresponds to a valid selection and vice versa, by showing a few ways to show that an object is decided in some direction. The first lemma is based on the candidate lists of the two guessed objects.

Lemma 6.4. *Let $\Phi := (x, y)$ be a guess and let $p \in X$ be an object such that $y \neq p \neq x$. If there exists a $c \in \{0, 1\}$ such that p is not in the candidate list of Φ_c, then p is decided in direction c.*

Proof. Let R be the initial arrangement of p and Φ_c. Let γ be a selection where

$$\gamma(p) \neq \gamma(\Phi_c).$$

We distinguish between two cases. In the first case the shared path $\xi_\gamma^R(p, \Phi_c)$ is empty. Then, the path $P_\gamma(p)$ of p and $P_\gamma(\Phi_c)$ of Φ_c are disjoint. Now consider a selection γ' where

$$\gamma(p) = \gamma(\Phi_c).$$

Because $P'_\gamma(p) \cup P_\gamma(p) = C_n$, we now know that

$$P'_\gamma(\Phi_c) \subset P'_\gamma(p)$$

and thus Φ_c shield p from its destination in γ'. Thus, a selection γ' where $\gamma'(p) = \gamma'(\Phi_c)$ is not valid.

In the second case the shared path $\xi_\gamma^R(p, \Phi_c)$ is non-empty. But then since p is not in the candidate list of Φ_c, there exists no edge on $\xi_\gamma^R(p, \Phi_c)$ where p and Φ_c can be swapped and thus p and Φ_c are not compatible in selection γ. Thus, a selection γ' where $\gamma'(p) \neq \gamma'(\Phi_c)$ is not valid. This concludes the proof. □

The second lemma focuses on the effect that decided objects have on other objects.

Lemma 6.5. *Let $(p, e) \in H$ such that $p \in \Phi$, i.e, p is a guessed object. If there exists an object $q \in C(p, e)$ such that q is decided in direction $1 - d(p, e)$, then for every $q' \in C(p, e)$ where $q' \neq q$ it holds that q' is decided in direction $d(p, e)$.*

Proof. We will prove the above statement by contradiction. Thus, suppose that there exists an undecided object $q' \in C(p, e)$. We will now show that if that is the case, then there exists no valid selection where q' is assigned to direction $1 - d(p, e)$ and thus, by definition, q' is decided, which leads to a contradiction.

Suppose there exists a valid selection γ that respects Φ such that $\gamma(q') = 1 - d(p, e)$. However, we also know that since q is decided in direction $1 - d(p, e)$, it holds for every valid selection γ' that $\gamma'(q) = 1 - d(p, e)$ and thus, also for γ. But then, by definition, $f_\gamma(p, e) > 1$ and thus, γ is not valid, a contradiction. □

Lastly, the third lemma focuses on objects that are in the end held by the agents that are neither on P_x nor on P_y.

Lemma 6.6. *Let $\mathcal{I} := ((N, X, \succ, C_n, \sigma_0, \sigma), e')$ be an instance of* First Swap Reachable Assignment *on cycle C_n with guess $\Phi := (x, y)$ and let x and y be non-opposite. Let q be an object such that the destination $\sigma(q)$ of q is not on the union of P_x and P_y. Then, for every $(p, e) \in H$ with $p \in \{x, y\}$, q is decided in direction $d(p, e)$.*

Proof. Since P_x and P_y only intersect at the edge e' where x and y are initially swapped according to the instance $\mathcal{I}$ of First Swap Reachable Assignment as defined above, there exists a direction $c \in \{0, 1\}$ such that q is not initially assigned to an agent on the path of guessed object $\Phi_c \in \{x, y\}$. But then, since, by assumption, the destination of q is also not on the path of Φ_c for both $c = 1$ and $c = 0$, for the initial arrangement R of q and Φ_c and every selection γ that respects Φ and where $\gamma(q) = c$, $\xi_\gamma^R(\Phi_c) \subset \xi_\gamma^R(q)$.

However, if that is true then, by definition, Φ_c shields q from its destination and thus, there exists no valid selection γ such that $\gamma(q) = c$. Therefore, q is decided in direction

$1-c$. We will now show that further there exists no $(p,e) \in H$ with $p \in \{x,y\}$, such that q is decided in direction $1-d(p,e)$. We make a case distinction over p.

In the first case $p = \Phi_{1-c}$ and it holds that $d(p,e) = 1-c$. However, since q is decided in direction $1-c$, q cannot be decided in direction $1-d(p,e)$.

In the second case $p = \Phi_c$. Since $\xi^R_\gamma(\Phi_c) \subset \xi^R_\gamma(q)$, in every γ' where $\gamma'(q) = 1-c$, $\xi^R_{\gamma'}(\Phi_c) \cap \xi^R_{\gamma'}(q) = \varnothing$. Therefore, q is not in any candidate list of Φ_c and hence q cannot be decided in direction $1-d(p,e)$. From this our statement follows. $\square$

6.2.2 Showing That the Number Of Undecided Objects in a Candidate List of a Guessed Object is Bounded by Two

We will proceed by formalizing the candidate lists of the guessed objects in Φ and further partition candidate lists with decided from candidate lists with undecided objects. Afterwards, we presents a sequence of auxiliary lemmas leading up to the proposition that the number of undecided objects in a candidate list of a guessed object is at most two. The following definition introduces various sets of objects and candidate lists that can be defined per guess Φ.

Definition 6.7. Let $\Phi := (x,y)$ be a guess. We partition the set of objects X into three subsets U, D and D_0 where U is the set of undecided objects, D is the set of decided objects that appear in at least one candidate list $C(p,e)$ of a guessed object $p \in \{x,y\}$ with direction $1-d(p,e)$ and where D_0 is the set of decided objects such that if an object $q \in D_0$ appears in candidate list $C(p,e)$ of a guessed object $p \in \{x,y\}$, then q is decided in direction $d(p,e)$.

Further, let $O \subseteq D$ denote the set of objects q for which there exists a guessed object $p \in \{x,y\}$ such that p and q are opposite and there exists an edge e such that $q \in C(p,e)$ and q is decided in direction $1-d(p,e)$.

Further, let $\mathcal{C}$ be the set of candidate lists $C(p,e)$ with $p \in \{x,y\}$. We partition the set $\mathcal{C}$ into two subsets $\mathcal{C}_D$ and $\mathcal{C}_U$ where $\mathcal{C}_D$ contains all candidate lists $C(p,e)$ where p is a guessed object and which contain an object $q \in D$ and where $\mathcal{C}_U$ contains all candidate lists $C(p,e)$ where p is a guessed object and which contains at least two distinct objects $q_0, q_1 \in U$.

First we state the following observation, based on Definition 6.7.

Observation 6.8. *Sets U, D and D_0 are a partition of X.*

We will further show under what condition $\mathcal{C}_D$ and $\mathcal{C}_U$ are a partition of $\mathcal{C}$.

Lemma 6.9. *Let $\Phi := (x,y)$ be a guess. Then $\mathcal{C}_D$ and $\mathcal{C}_U$ are a partition of $\mathcal{C}$ or Φ is wrong.*

Proof. We prove the above statement in two steps. First we prove that $\mathcal{C}_D \cap \mathcal{C}_U = \varnothing$. Afterwards we prove that $\mathcal{C}_D \cup \mathcal{C}_U = \mathcal{C}$ or the target assignment σ cannot be reached with guess Φ. As a result, $\mathcal{C}_D$ and $\mathcal{C}_U$ are partition of $\mathcal{C}$ or the target assignment σ cannot be reached with guess Φ.

According to Lemma 6.5, a candidate list $C(p, e)$ cannot contain two objects q_0 and q_1 such that q_0 is decided in direction $1-d(p, e)$ and q_1 is undecided. Therefore $\mathcal{C}_D \cap \mathcal{C}_U = \emptyset$.

We will now prove that $\mathcal{C}_D \cup \mathcal{C}_U = \mathcal{C}$ or the target assignment σ cannot be reached with guess Φ. Suppose there exists a candidate list $C(p, e)$ such that $C(p, e) \notin \mathcal{C}_D$ and $C(p, e) \notin \mathcal{C}_U$. But then, $C(p, e)$ neither contains undecided nor objects q such that q is decided in direction $1 - d(p, e)$. But then, by definition, $C(p, e)$ contains only objects in direction $d(p, e)$. However, that means in every valid selection γ, for every object q in $C(p, e)$ it holds that $\gamma(q) = d(p, e)$. Further, since p is a guessed object, $\gamma(p) = d(p, e)$. Thus, $f_\gamma(p, e) = 0$ and γ is not valid, a contradiction. Since consequentially there exists no valid selection, σ cannot be reached with guess Φ. Suppose no such $C(p, e)$ exists. Then $\mathcal{C}_D \cup \mathcal{C}_U = \mathcal{C}$.

Thus, $\mathcal{C}_D$ and $\mathcal{C}_U$ are partition of $\mathcal{C}$ or the target assignment σ cannot be reached with guess Φ. □

The following two lemmas are auxiliary lemmas for proving that the number of undecided objects in a candidate list of a guessed object is at most two. The next lemma shows that every undecided objects is in exactly two candidate lists in $\mathcal{C}$.

Lemma 6.10. *Let Φ be a guess. For every undecided object q, there exist exactly two candidate lists $C_0, C_1 \in \mathcal{C}$ such that $q \in C_0$ and $q \in C_1$ or the target assignment σ cannot be reached with guess Φ.*

Proof. Suppose the statement is not true. Then either q appears only in one candidate list and q is decided due to Lemma 6.4 or there exist three candidate lists $C_0, C_1, C_2 \in \mathcal{C}$ such that $q \in C_0, q \in C_1$ and $q \in C_2$. But then there must exist a guessed object $p \in \{x, y\}$ such that q is in two candidate lists of p. We make the following case distinction.

In the first case, q and p are opposite. However, since the direction of the guessed object p is fixed and hence, there exists no valid selection γ where $\gamma(q) = \gamma(p)$, q is decided, a contradiction.

In the second case q and p can be swapped twice on the same shared path but then, due to Lemma 3.7, the target assignment σ cannot be reached with guess Φ and hence, Φ is wrong. □

The following lemma shows a property of the guessed objects, given that there exist objects opposite to a guessed object whose decided paths intersect with the guessed object's, i.e. $|O| > 0$.

Lemma 6.11. *Let $\Phi := (x, y)$ be a guess. If $|O| > 0$, then the guessed objects x and y are opposite or the guess Φ is wrong.*

Proof. Let q be an object in O. That means that there exists a candidate list $C(p, e) \in \mathcal{C}$ such that $q \in C(p, e)$, q is decided in direction $1 - d(p, e)$ and q is opposite to p. Let r be the other guessed object. We will now show that p and r must be opposite or the guess Φ is wrong. Note that since q is decided in direction $1-d(p, e)$ it must be swapped with q at e. However, before p meets q, it is swapped with r in the first swap. Suppose that p and r are not opposite. Then p and r are only swapped once. However, since q is assigned the same direction as r and r reaches its destination before it meets p a second time, also q cannot meet p a second time and thus, there exists no valid selection that respects this guess and hence, the guess is wrong. □

We will now prove the following equality regarding the cardinality of $\mathcal{C}$.

Lemma 6.12. *Let $\Phi := (x, y)$ be a guess. If x and y are opposite, then*

$$|\mathcal{C}| = n + |O|$$

or Φ is wrong.

Proof. Let $p \in \{x, y\}$ be a guessed object and let $r \in \{x, y\}$ with $p \neq r$ be the other guessed object. Suppose that p and r are opposite. We partition the set O into two subsets, Q_p and Q_r, as follows: Let q be an object such that there exists a candidate list $C(p, e) \in \mathcal{C}$ such that $q \in C(p, e)$, q is decided in direction $1 - d(p, e)$ and q is opposite to p. Then $q \in Q_p$. Otherwise q, since it is in O, is in some candidate list $C(r, e)$ and decided in direction $1 - d(r, e)$. Then $q \in Q_r$.

Observe that p is swapped with r before it is swapped with any object in O because p and r are the guessed objects in Φ. Since p and r are opposite, they need to be swapped twice as otherwise the guess is wrong. Suppose the guess is not wrong. The union $P_x \cup P_y$ covers the whole cycle C_n and hence, there exist at least n candidate lists, i.e. $|\mathcal{C}| \geqslant n$. Note that after p is swapped with r it is swapped with every object in O_p once, before reaching its destination. Since p and r are symmetric, we can derive the following. The guessed object p is swapped with $|O_p|$ objects after it is swapped with r the second time and guessed object r is swapped with $|O_r|$ objects after it is swapped with p the second time. From this it follows that the number of candidate lists is equal to $n + |O_p| + |O_r| = n + |O|$. This concludes the proof. □

Before we come to the prove that the number of undecided objects in a candidate list of a guessed object is at most two, we will first show in the following two lemmas how the cardinalities of the sets $\mathcal{C}_U$ and U are related. In the next lemma we will show an equality involving the cardinality of $\mathcal{C}_D$ as well as the sets D and O. We will use this result later in our proposition that the number of undecided objects in a candidate list of a guessed object is upper-bounded by two.

Lemma 6.13. *Let $\Phi := (x, y)$ be a guess. Then either Φ is wrong or it holds that*

$$|\mathcal{C}_D| = |D| + |O|. \tag{6.3}$$

Proof. We start with the following observation which is true due to $O \subseteq D$:

$$|D \backslash O| = |D| - |O|.$$

First observe that every candidate list $C(p, e) \in \mathcal{C}_D$ contains exactly one object $q \in D$ such that q is decided in direction $1 - d(p, e)$, due to Lemma 6.5.

Now recall that every object $q \in D \backslash O$ is non-opposite to both guessed objects x and y and can therefore appear in at most one candidate list of x and in one candidate list of y, or otherwise Φ is wrong. Further, since $q \in D \backslash O$, object q is decided. However, since in every valid selection that respect Φ, $\gamma(x) \neq \gamma(y)$, q is either decided in direction $1 - d(x, e_x)$ for some edge e_x such that $q \in C(x, e_x)$ or q is decided in direction $1 - d(y, e_y)$ for some edge e_y such that $q \in C(y, e_y)$ but not both. Thus, the number ϵ_0 of candidate lists $C(p, e)$ that contain an edge $q \in D \backslash O$ is equal to the number of objects in $D \backslash O$.

Next, recall that every object $q \in O$ is opposite to one guessed object $p \in \{x, y\}$ for which there exists an edge e such that $q \in C(p, e)$ and q is decided in direction $1 - d(p, e)$. Hence q appears in two candidate list of p and in no candidate list of the other guessed object $r \neq p$, or otherwise Φ is wrong. Thus, the number ϵ_1 of candidate lists $C(p, e)$ that contain an edge $q \in O$ is equal to two times the number of objects in O.

We will now combine these results into Equation 6.3. Observe from how we calculated ϵ_0 and ϵ_1, that:

$$|\mathcal{C}_D| = \epsilon_0 + \epsilon_1.$$

From this we can derive the following equation:

$$|\mathcal{C}_D| = \epsilon_0 + \epsilon_1 = |D \backslash O| + 2|O|$$

which can be rewritten as

$$|\mathcal{C}_D| = |D| - |O| + 2|O| = |D| + |O|.$$

□

The following two lemmas show an inequality regarding the cardinality of $\mathcal{C}_U$. The first lemma shows this inequality for the case that x and y are opposite.

Lemma 6.14. *Let $\Phi := (x, y)$ be a guess where x and y are opposite. Then,*

$$|\mathcal{C}_U| \geqslant |U| \tag{6.4}$$

or Φ is wrong.

Proof. Since x and y are opposite, due to Lemma 6.12, the following holds:

$$|\mathcal{C}| = n + |O|$$

where n is the number of objects. Recall that $\mathcal{C}_D$ and $\mathcal{C}_U$ are a partition of $\mathcal{C}$ and thus,

$$|\mathcal{C}_D| + |\mathcal{C}_U| = n + |O|.$$

Substituting by Equation 6.3, whose correctness we proved in Lemma 6.13 yields:

$$|D| + |O| + |\mathcal{C}_U| = n + |O|.$$

Further, rearranging the terms yields:

$$|\mathcal{C}_U| = n - |D|.$$

Note that $n = |U| + |D| + |D_0|$, since U, D and D_0 are a partition of X and hence, the following inequality holds:

$$|\mathcal{C}_U| \geqslant |U|.$$

This concludes our proof. □

In the next lemma we will show the relationship between the cardinalities of $\mathcal{C}_U$ and U, given that x and y are non-opposite.

Lemma 6.15. *Let $\Phi := (x, y)$ be a guess where x and y are non-opposite. Then,*

$$|\mathcal{C}_U| \geqslant |U| \tag{6.5}$$

or Φ is wrong.

Proof. We prove the above statement in two steps. In the first step we will show that if x and y are non-opposite, then we can derive from the result of Lemma 6.6, that either Φ is wrong or the following holds:

$$|\mathcal{C}| \geqslant |U| + |D|.$$

Afterwards we derive the above statement from that result.

We will now prove the first step. Recall the result of Lemma 6.6, which states that if x and y are opposite, then every object whose destination is not on the joint path $P_x \cup P_y$ of x and y is in D_0. Hence, if an object is in U or D, then its destination is on $P_x \cup P_y$. Note further that for an object there exists as many candidate lists as there are edges on that objects path. Thus, the number objects in U and D are upper-bounded by the number of candidate lists of x and y. From this it directly follows that

$$|\mathcal{C}| \geqslant |U| + |D|.$$

We will now show the second step. Since $\mathcal{C}_D$ and $\mathcal{C}_U$ are a partition of $\mathcal{C}$, we can write the above equation as follows:

$$|\mathcal{C}_D| + |\mathcal{C}_U| \geqslant |U| + |D|.$$

Substituting $|\mathcal{C}_D|$ according to Equation 6.3, whose correctness was shown in Lemma 6.13, yields:

$$|D| + |O| + |\mathcal{C}_U| \geqslant |U| + |D|.$$

Now recall that, due to Lemma 6.11, since x and y are non-opposite, $|O| = 0$ or Φ is wrong. Setting $|O| = 0$ and subtracting both sides by $|D|$ yields:

$$|\mathcal{C}_U| \geqslant |U|.$$

This concludes our proof. □

We will now conclude this section by showing that the number of undecided objects in a candidate list of the guessed objects x and y is upper-bounded by two.

Proposition 6.16. *Let $\Phi := (x, y)$ be a guess. Let further $(p, e) \in H$, such that $p \in \Phi$, i.e, p is a guessed object. The number of undecided objects in $C(p, e)$ is upper-bounded by two or Φ is wrong.*

Proof. We will prove the above statement by using a counting argument. We have already shown in Lemma 6.14 and Lemma 6.15 that it holds

$$|\mathcal{C}_u| \geqslant |U|$$

or Φ is wrong. Further in Lemma 6.10 we showed that every undecided object is in exactly two candidate lists $C(p, e)$ and $C(q, e)$ where $p, q \in \{x, y\}$. Lastly, by definition of $\mathcal{C}_U$ every candidate list of $C(p, e) \in \mathcal{C}_U$ contains at least two disjoint objects $q_0, q_1 \in U$.

From this it follows that every candidate list $C(p, e) \in \mathcal{C}_U$ contains exactly two undecided objects, as otherwise there exist not enough undecided objects to fill every candidate list in $\mathcal{C}_U$ with at least two undecided objects, since every undecided object can be in at most two candidate lists in $\mathcal{C}_U$.

Now, since every candidate list in $\mathcal{C}_U$ contains exactly two objects and since every candidate list in $\mathcal{C}_D$ contains zero undecided objects, the number of undecided objects in any candidate list in $\mathcal{C}$ is upper-bounded by two. □

6.2.3 Simulating the Edge Where Two Objects Meet Given A Direction Assignment

In this section we will first show that, given an instance of Reachable Assignment we can compute the number of objects moving into clockwise direction in every valid selection. Afterwards, we will use that intermediate result to prove that if a selection is valid, then, given two objects with a shared path P, we can compute the edge e on P where the two objects meet in every swap sequence that transforms σ_0 into σ. This result will then allow us to construct a 2-SAT formula such that every satisfying truth assignment corresponds to a valid selection and vice versa in Section 6.3.

Lemma 6.17. *Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be a instance of* Reachable Assignment *on cycle C_n. For the set of objects X one can calculate the number of objects moving in clockwise and respectively counter-clockwise direction in every valid selection.*

Proof. Let the variables $d_1, ..., d_n \in \{0, 1\}$ represent objects $x_1, ..., x_n$ where x_i moves in clockwise direction if $d_i = 1$ and counter-clockwise otherwise. Let further $y_1, ..., y_n \in \mathbb{N}$ denote the length of the path of an object's initial position to its destination in clockwise direction. Since for each edge on an object's path in a certain direction there must be an object moving in the opposite direction swapping the object at that edge the sum of the lengths of the paths of the objects moving in clockwise direction must be equal to the sum of the lengths of the paths of the objects moving in counter-clockwise direction. So, formally, it must hold:

$$\sum_{i=1}^{n} d_i y_i = \sum_{i=1}^{n} ((1 - d_i)(n - y_i)) \tag{6.6}$$

This can be rewritten as:

$$\sum_{i=1}^{n}(d_i y_i - (1-d_i)(n-y_i)) = 0$$

Expanding the terms and reordering them yields:

$$\sum_{i=1}^{n}(d_i y_i - (n - nd_i - y_i + d_i y_i)) = \sum_{i=1}^{n}(d_i y_i - n + nd_i + y_i - d_i y_i)$$

$$= \sum_{i=1}^{n}(-n + nd_i + y_i) = -\sum_{i=1}^{n} n + n\sum_{i=1}^{n} d_i + \sum_{i=1}^{n} y_i = 0$$

$\sum_{i=1}^{n} n$ is simply n^2 and $Y = \sum_{i=1}^{n} y_i$ is a constant given the graph, the objects and the assignments σ_0 and σ. Then the equation can be rewritten as:

$$\sum_{i=1}^{n} d_i = \frac{n^2 - Y}{n} = \frac{n^2}{n} - \frac{Y}{n} = n - \frac{Y}{n} \tag{6.7}$$

Hence, the number of objects moving in clockwise direction is equal to $n - \frac{Y}{n}$ where $\frac{Y}{n}$ denotes the number of objects moving in counter-clockwise direction. □

We will henceforth denote the number of objects walking counter-clockwise direction in instance $\mathcal{I}$ by $\theta_{\mathcal{I}}$ or θ for short if the instance is implicitly clear. Given some direction $d \in \{0,1\}$ the following term is equal to the number of objects walking in that direction in every valid selection:

$$dn + (-1)^d \theta \tag{6.8}$$

which is equal to θ for $d = 0$ and $n - \theta$ for $d = 1$. We will now use this result to prove the following statement which we will use to derive an important concept that will be used later to construct a 2-SAT formula such that every satisfying truth assignment of that formula corresponds to a valid selection and vice versa.

Lemma 6.18. *Let* $\mathcal{I}_i := ((N, X, \succ, C_n, \sigma_0, \sigma), e')$ *be an instance of* First Swap Reachable Assignment *on cycle* C_n*. Let* $\Phi := (x, y)$ *be the guess of* $\mathcal{I}_i$*, let* $(p, e) \in H$ *be an object-edge-pair and let* $q \in C(p, e)$ *where* p *and* q *are undecided objects. Let* P *be the shared path of* p *and* q *such that* $e \in E(P)$*. Then, there exists an edge* f *on* P *such that for every valid selection* γ *that respects* Φ *and for which its holds that*

$$\gamma(p) = d(p, e) = 1 - \gamma(q),$$

p and q meet at edge f in the execution of Algorithm 1 on input γ.

Proof. Let $\Phi := (x, y)$ be the guess of $\mathcal{I}_i$. In the following proof we will assume that $\gamma(p) = \gamma(x)$. The other case is symmetric with x and y and clockwise and counter-clockwise swapped, without loss of generality.

We will first show that given the object p we can compute the constant number $\lambda(p, x)$ of objects on the object domain $\Delta_{p,x}$ that are assigned clockwise direction in every valid selection γ where $\gamma(p) = \gamma(x)$. Afterwards we use that result and show, due to the fact that we know that in every valid selection γ it holds that $\gamma(x) = 1$, that $\mathcal{S}_\gamma$, as defined above, exists.

We will now prove that for any valid selection γ where $\gamma(p) = \gamma(x)$, $\lambda(p, x)$ is equal to the number of edges that y has to pass before reaching the edge e_y where p and y can be swapped. Since p is undecided and hence in the candidate list of y we know that e_y must exist and that p and y have to be swapped at e_y or otherwise the target assignment σ cannot be reached. Further we know that y has to be swapped with $\lambda(p, x)$ objects (including x) before it meets p, as these are the objects on the swap space of p and y that are assigned the opposite direction of y. Suppose that $\lambda(p, x)$ is not equal to the number of edges that y has to pass before reaching e_y but γ is valid. Because y has to be swapped with $\lambda(p, x)$ objects and each swap amounts to one edge being passed, y and p meet at an edge f that is not e_y. However, Lemma 3.7 states that there exists exactly one edge on the shared path of p and y in their initial arrangement where p and y can be swapped, which is e_y. Thus, p and y cannot be swapped at edge f and σ cannot be reached and therefore γ cannot be valid, a contradiction. Since edge e_y is constant for all valid selections, so is $\lambda(p, x)$.

We will now extend this result by proving that for every valid selection γ where $\gamma(p) = \gamma(x)$, the number $\lambda(x, p)$ of objects on the object domain $\Delta_{x,p}$ that are assigned to clockwise direction is constant. Recall Equation 2.5 with adjusted variable names:

$$\forall r, t \in X. \Delta_{r,t} \cup \Delta_{t,r} = X.$$

The equation states that the union of the object domain from an object r in clockwise direction to t and the object domain from an object r in counter-clockwise direction to t covers every object in instance $\mathcal{I}_i$. So to compute $\lambda(x, p)$ we simply compute $\lambda(p, x)$ and subtract it from the total number of objects walking in clockwise direction, that is

$$n - \theta_{\mathcal{I}} \tag{6.9}$$

where $\mathcal{I}$ is the original instance of Reachable Assignment of $\mathcal{I}_i$. Further since x and p already walk in clockwise direction we need to subtract 1 for each of them, to not count them twice, and obtain the following result.

$$\lambda(x, p) = n - \theta_{\mathcal{I}} - \lambda(p, x) - 2. \tag{6.10}$$

Since $\lambda(p, x)$ and $n - \theta_{\mathcal{I}}$ are constants, so is $\lambda(x, p)$.

We will now show that for an object q in opposite direction to p, where the shared path of p and q is non-empty, we can compute the edge e_p where p and q meet on that shared path, in every valid selection γ.

Since we assumed $\gamma(p) = 1$, the direction of q is counter-clockwise. We distinguish between two cases. In the first case, q is closer to p than to x with respect to selection γ. For γ to be valid, we know that, since q is undecided and hence in the candidate list of x, q and x have to be swapped at some edge e_x. We further know how many objects q is swapped with after it is swapped with p and before it can be swapped with x, namely $\lambda(x, p)$. We can therefore calculate the edge e_p, where p and q have to meet and swap, such that q can meet x at edge e_x. In the second case x is closer to q than p with respect to selection γ. For γ to be valid, q and x have to be swapped at some edge e_x. We further know how many objects q is swapped with after it is swapped with x and before it can be swapped with p, namely $\lambda(p, x)$. We can therefore calculate the edge e_p, where p and q will meet after q was swapped with x at edge e_x. Since $\lambda(p, x)$ and $\lambda(x, p)$ are constant for all valid selections, so it the edge e_p. □

We formalize the results of lemma 6.18 as follows.

Definition 6.19. Let $(p, e) \in H$ be an object-edge-pair and let $q \in C(p, e)$. Let P be the shared path of p and q such that $e \in E(P)$. According to Lemma 6.18, there exists an edge f on P such that for every valid selection γ that respects Φ and for which it holds that

$$\gamma(p) = d(p, e) = 1 - \gamma(q),$$

p and q meet at edge f in the execution of Algorithm 1 on input γ. If q is closer to p in direction $d(p, e)$ than guessed object $\Phi_{d(p,e)}$, then let $c := 1$ and $c := 0$ otherwise. We denote by $S(p, e, q)$ the tuple (f, c). If $e = f$, we say that $S(p, e, q)$ is successful and we say that $S(p, e, q)$ is unsuccessful otherwise.

Based on this definition, we will prove the following two statements that we will use as an intermediate result for the construction of a 2-SAT formula such that every satisfying truth assignment of that formula corresponds to a valid selection and vice versa.

Lemma 6.20. *Let p, q_0 and q_1 be three objects and let e be an edge. If $S(p, e, q_0) = S(p, e, q_1)$, then q_0 and q_1 are in the same candidate list of guessed object $\Phi_{d(p,e)}$.*

Proof. Let e_0 be the edge such that for every valid selection γ that respects Φ and for which its holds that

$$\gamma(p) = d(p, e) = 1 - \gamma(q_0),$$

p and q_0 meet at edge e_0 in the execution of Algorithm 1 on input γ, according to Lemma 6.18. Let e_1 be the edge such that for every valid selection γ that respects Φ and for which its holds that

$$\gamma(p) = d(p, e) = 1 - \gamma(q_1),$$

p and q_1 meet at edge e_1 in the execution of Algorithm 1 on input γ, according to Lemma 6.18. By definition, if $S(p, e, q_0) = S(p, e, q_1)$ then the following statements are true.

1. Objects q_0 and q_1 are either both closer to p in direction $d(p, e)$ than the guessed object $\Phi_{d(p,e)}$ or both further from p in direction $d(p, e)$ than $\Phi_{d(p,e)}$.

2. $e_0 = e_1$.

Let $(f, c) = S(p, e, q_0) = S(p, e, q_1)$. We distinguish between two cases. In the first case $c = 1$. Then, according to Lemma 6.18, there exists a number λ of objects such that q_0 and q_1 are both swapped with exactly λ objects after being swapped with p and before meeting guessed object $\Phi_{d(p,e)}$, at an edge e_0^* and e_1^* respectively.

In the second case $c = 0$. Then, according to Lemma 6.18, there exists a number λ of objects such that q_0 and q_1 are both swapped with exactly λ objects after being swapped with guessed object $\Phi_{d(p,e)}$, at an edge e_0^* and e_1^* respectively, and before meeting p.

But in both cases, since λ is the same for both q_0 and q_1, it holds that $e_0^* = e_1^*$. In Lemma 6.18, e_0^* and e_1^* are chosen to be the edges where $\Phi_{d(p,e)}$ can be swapped with q_0 and q_1 respectively. Thus, q_0 and q_1 are in the same candidate list of $\Phi_{d(p,e)}$. □

We conclude this section with the second statement that we can derive from Lemma 6.18. We will use both of these two Lemmas to construct a 2-SAT formula such that every selection is valid if and only if it corresponds to a satisfying truth assignment of that formula.

Lemma 6.21. *Let $\Phi := (x, y)$ be the guess of an instance $\mathcal{I} := ((N, X, \succ, C_n, \sigma_0, \sigma), e')$ of* First Swap Reachable Assignment *on cycle C_n, let $(p, e) \in H$ and let $q \in C(p, e)$. If $S(p, e, q)$ is not successful, then there exists no valid selection γ such that $\gamma(p) = d(p, e) = 1 - \gamma(q)$.*

Proof. Let P be the shared path of p and q such that $e \in E(P)$. If $S(p, e, q)$ is not successful, then, according to Lemma 6.18, there exists an edge f on P such that for every valid selection γ that respects Φ and for which its holds that

$$\gamma(p) = d(p, e) = 1 - \gamma(q),$$

p and q meet at edge f in the execution of Algorithm 1 on input γ and $e \neq f$.

Recall Lemma 3.7. According to this lemma, since f and e are on the same shared path of p and q, there exists at most one edge of e and f where p and q can be swapped or otherwise σ is not reachable. But if γ is valid, then Algorithm 1 finds a swap sequence that reaches σ. Since $q \in C(p, e)$, it is guaranteed that p and q can be swapped at e and therefore p and q cannot be swapped at f. Thus, there cannot exist a valid selection γ that respects Φ and for which its holds that

$$\gamma(p) = d(p, e) = 1 - \gamma(q).$$

□

6.3 Constructing a 2-SAT formula of Constraints with More Than Two Objects Involved

In this section we will construct a 2-SAT formula for First Swap Reachable Assignment on cycles such that every selection is valid if and only if it corresponds to a satisfying truth assignment of that formula. We will then combine the results of n such formulas to solve Reachable Assignment on cycles.

We will discuss the construction of ϕ in multiple steps and we distinguish between five categories of objects. First we create two formulas for decided and undecided objects in general. Then we create a 2-SAT formula for all pairwise opposite objects. Afterwards we will create a 2-SAT formula for all objects for which there exists an edge where the corresponding candidate list of that edge contains at least one decided object. Lastly, we create a 2-SAT formula for all objects that are not covered by one of the previous four categories. The conjunction of the resulting five formulas together with the 2-SAT formula ψ, for which every satisfying truth assignment corresponds to a harmonic selection and vice versa, will then be a 2-SAT formula such that a selection is valid if and only if it corresponds to a satisfying truth assignment of that formula.

We start by partitioning the set H into five sets. Successively we will propose a 2-SAT formula for every of those sets and afterwards show that the conjunction of the five resulting formulas solves every instance of First Swap Reachable Assignment on cycles. We start by introducing the following notation.

Definition 6.22. Let $\Phi := (x, y)$ be a guess. Let $(p, e) \in H$ be an object-edge-pair. Then $\eta_o(p, e)$ denotes the number of objects $q \in C(p, e)$ that are opposite to p. Further $\eta_d(p, e)$ denotes the number of objects $q \in C(p, e)$ that are decided in direction $d(p, e)$ and $\eta_{1-d}(p, e)$ denotes the number of objects $q \in C(p, e)$ that are decided in direction $1 - d(p, e)$.

Using this notation, we partition the set H into four sets as follows.

Definition 6.23. Let $\Phi := (x, y)$ be a guess. We will partition the set H into four partitions. Recall how we constructed $\mathcal{C}$ as the set of candidate lists $C(p, e)$ such that $p \in \{x, y\}$. Analogously we define $H(\mathcal{C})$ as follows:

$$H(\mathcal{C}) := \{(p, e) \in H \mid p \in \{x, y\}\}.$$

Further we define the sets $H(\mathcal{C}_D)$ and $H(\mathcal{C}_U)$ as follows:

$$H(\mathcal{C}_D) := \{(p, e) \in H(\mathcal{C}) \mid \exists q \in C(p, e) \textit{ such that } q \textit{ is decided in direction } 1 - d(p, e)\},$$

$$H(\mathcal{C}_U) := \{(p, e) \in H(\mathcal{C}) \mid \exists q_0, q_1 \in C(p, e) \textit{ such that } q_0 \textit{ and } q_1 \textit{ are undecided}\}.$$

Further we define the set H_0 as follows:

$$H_0 := \{(p, e) \in H \backslash H(\mathcal{C}) \mid \eta_o(p, e) > 0\}.$$

Further we define the set H_1 as follows:

$$H_1 := \{(p, e) \in H \backslash (H_0 \cup H(\mathcal{C})) \mid \eta_d(p, e) = |C(p, e)| \textit{ or } \eta_{1-d}(p, e) > 0\}.$$

Lastly we define the set H_2 as follows:

$$H_2 := H \backslash (H_0 \cup H_1 \cup H(\mathcal{C})).$$

Observation 6.24. *The sets $H(\mathcal{C}), H_0, H_1$ and H_2 are a partition of H and $H(\mathcal{C}_D)$ and $H(\mathcal{C}_U)$ are a partition of $H(\mathcal{C})$.*

The following observation will assist us to discuss the time complexity of our constructions. Afterwards we begin with construction the remaining 2-SAT formulas.

Observation 6.25. *Let $p \in X$ be an object. Due to Lemma 3.9 an object $q \in X$ where $p \neq q$ can be in at most two candidate lists of p. Therefore, the following equation holds:*

$$\sum_{e \in E} |C(p,e)| \leqslant 2|X|.$$

For the subsequent constructions recall the function f_γ for a selection γ as defined in Definition 5.4. The following construction formalizes a 2-SAT formula that assigns to every decided object the direction in which the object is decided and thereby constructs the 2-SAT formula f_D for all objects in $H(\mathcal{C}_D)$.

Construction 6.26. Let $\Phi := (x,y)$ be the guess of an instance $\mathcal{I} := ((N, X, \succ, C_n, \sigma_0, \sigma), e')$ of First Swap Reachable Assignment on cycle C_n. Let $C(p,e) \in \mathcal{C}_D$ be a candidate list and let $q \in C(p,e)$ be the object such that q is decided in direction $1 - d(p,e)$. Let then $Q := C(p,e) \backslash \{q\}$ be the set of decided objects in direction $d(p,e)$. We construct f_D as the conjunction of

$$\begin{cases} \neg q & \textit{if } d(p,e) = 1 \\ q & \textit{otherwise} \end{cases} \tag{6.11}$$

and

$$\begin{cases} \bigwedge_{q' \in Q} q' & \textit{if } d(p,e) = 1 \\ \bigwedge_{q' \in Q} \neg q' & \textit{otherwise.} \end{cases} \tag{6.12}$$

We will now prove that the construction of f_D is correct.

Lemma 6.27. *Let $\Phi := (x,y)$ be a guess and let $C(p,e) \in \mathcal{C}_D$ be a candidate list. Suppose there exists a valid selection for guess Φ. Let γ be a selection that respects Φ. If γ is valid, then it corresponds to a satisfying truth assignment of f_D and if γ corresponds to a satisfying truth assignment of f_D, then it holds that $f_\gamma(p,e) = 1$.*

Proof. We will prove both directions of this statement by contradiction.

First suppose that γ is valid but it does not correspond to a satisfying truth assignment of f_D. Then, there exists a clause c in f_D that is not fulfilled by γ. We distinguish between two cases.

In the first case c is a clause as in Equation 6.11. But then there exists a decided object q in direction $1 - d(p,e)$ such that $\gamma(q) = d(p,e)$. But by definition of decided objects, there exists no valid selection γ' such that $\gamma'(q) = d(p,e)$ and thus, γ is not valid, a contradiction.

In the second case c is a clause as in Equation 6.12. But then there exists a decided object q' in direction $d(p,e)$ such that $\gamma(q') = 1 - d(p,e)$. But by definition of decided objects, there exists no valid selection γ' such that $\gamma'(q') = 1 - d(p,e)$ and thus, γ is not valid, a contradiction.

We will now prove the other direction of the statement. Suppose that γ corresponds to a satisfying truth assignment of f_D but $f_\gamma(p,e) \neq 1$. But if γ corresponds to a satisfying truth assignment of f_D, then there exists one object $q \in C(p,e)$ that is assigned to direction $1 - d(p,e)$ in Equation 6.11 and further, due to Equation 6.12, no other object in $C(p,e)$. Thus, $f_\gamma(p,e) = 1$, a contradiction. □

We continue by constructing a 2-SAT formula that ensures that every pair of undecided objects in the same candidate list of a guessed object are never assigned the same direction. We will prove afterwards that this is correct. We construct the 2-SAT formula f_U for all objects in $H(\mathcal{C}_U)$ as follows.

Construction 6.28. Let $\Phi := (x,y)$ be the guess of an instance $\mathcal{I} := ((N, X, \succ, C_n, \sigma_0, \sigma), e')$ of First Swap Reachable Assignment on cycle C_n.

Let $C(p,e) \in \mathcal{C}_U$ be a candidate list and let $q_0, q_1 \in C(p,e)$ be the two undecided objects in $C(p,e)$. Then, add to f_U the term

$$q_0 \neq q_1. \tag{6.13}$$

We will now show that the construction of f_U is correct.

Lemma 6.29. *Let $\Phi := (x,y)$ be a guess and let $C(p,e) \in \mathcal{C}_U$ be a candidate list. Suppose there exists a valid selection for guess Φ. Let γ be a selection that respects Φ. If γ is valid, then it corresponds to a satisfying truth assignment of f_U and if γ corresponds to a satisfying truth assignment of f_U, then it holds that $f_\gamma(p,e) = 1$.*

Proof. We will prove both directions of this statement by contradiction.

First suppose that γ is valid but it does not correspond to a satisfying truth assignment of f_U. Then there exists a clause c in f_U such that c is not fulfilled in γ. Then, this clause is the clause in Equation 6.13 and there exist two undecided objects $q_0, q_1 \in C(p,e)$ such that $\gamma(q_0) = \gamma(q_1)$. However, since due to Proposition 6.16, there exists exactly two undecided objects in $C(p,e)$, either $f_\gamma(p,e) = 2$, if $\gamma(q_0) = \gamma(q_1) = 1 - d(p,e)$ or otherwise $f_\gamma(p,e) = 0$, if $\gamma(q_0) = \gamma(q_1) = d(p,e)$. In both cases, $f_\gamma(p,e) \neq 1$ and thus, γ is not valid, a contradiction.

We will now show the other direction of the statement. Suppose that γ corresponds to a satisfying truth assignment of f_U but $f_\gamma(p,e) \neq 1$. However, since due to Proposition 6.16, there exists exactly two undecided objects q_0, q_1 in $C(p,e)$ and since γ corresponds to a satisfying truth assignment of f_U, $\gamma(q_0) \neq \gamma(q_1)$ and thus $f_\gamma(p,e) = 1$, a contradiction. □

In the next step we construct a 2-SAT formula φ_0 for all objects in H_0.

Construction 6.30. Let $\Phi := (x, y)$ be the guess of an instance $\mathcal{I} := ((N, X, \succ, C_n, \sigma_0, \sigma), e')$ of First Swap Reachable Assignment on cycle C_n. We will now construct a 2-SAT formula f_0 with parameter (p, e) as follows. Since $(p, e) \in H_0$, $\eta_o(p, e) > 0$. We distinguish between two cases.

In the first case $\eta_o(p, e) > 1$. Then, add to $f_0(p, e)$ the term

$$\begin{cases} \neg p & \text{if } d(p, e) = 1 \\ p & \text{otherwise.} \end{cases} \tag{6.14}$$

In the second case $\eta_o(p, e) = 1$. Then let $q \in C(p, e)$ be the object in the candidate list of p at e that is opposite to p. Let $Q := C(p, e) \backslash \{q\}$. Then, add to $f_0(p, e)$ the term

$$\begin{cases} \bigwedge_{q' \in Q} (p \to q') & \text{if } d(p, e) = 1 \\ \bigwedge_{q' \in Q} (\neg p \to \neg q') & \text{otherwise.} \end{cases} \tag{6.15}$$

We further define φ_0 as follows:

$$\varphi_0 := (\bigwedge_{(p,e) \in H_0} f_0(p, e)) \wedge \psi.$$

We will now show that the construction of φ_0 is correct.

Lemma 6.31. *Let $\Phi := (x, y)$ be a guess and let $(p, e) \in H_0$. Suppose there exists a valid selection for guess Φ. Let γ be a selection that respects Φ. If γ is valid, then it corresponds to a satisfying truth assignment of φ_0 and if γ corresponds to a satisfying truth assignment of $\varphi_0(p, e)$, then $f_\gamma(p, e) = 1$.*

Proof. We will show both directions of this statement by contradiction.

First suppose that γ is valid but it does not correspond to a satisfying truth assignment of φ_0. Then there exists an object-edge-pair $(p', e') \in H_0$ and a clause c in $f_0(p', e')$ such that c is not fulfilled in γ. We distinguish between two cases.

In the first case c is a term as defined in Equation 6.14. But then, there exist at least two objects q_0, q_1 that are opposite to p' and since γ is valid it holds that $\gamma(q_0) = \gamma(q_1) \neq \gamma(p')$ but since c is not fulfilled by a truth assignment corresponding to γ, $\gamma(p') = d(p', e')$. However, then $\gamma(q_0) = \gamma(q_1) = 1 - d(p, e)$ and thus, $f_\gamma(p', e') \geqslant 2$ and γ is not valid, a contradiction.

In the second case c is a term as defined in Equation 6.15. Then either the term in Equation 6.14 is also not fulfilled, and as shown, γ is not valid, a contradiction, or $\gamma(p') = d(p', e')$ and there exist at least two objects q_0, q_1 such that $\gamma(q_0) = \gamma(q_1) = 1 - d(p', e')$ and thus, $f_\gamma(p', e') \geqslant 2$. Then, however, γ is not valid, a contradiction.

We will now show the other direction of the statement. Suppose that γ corresponds to a satisfying truth assignment of f_0 but $f_\gamma(p, e) \neq 1$. Since $(p, e) \in H_0$, it holds that $\eta_o(p, e) > 0$. We distinguish between two cases.

In the first case $\eta_o(p, e) > 1$, then $\gamma(p) = 1 - d(p, e)$, according to Equation 6.14. But then, $f_\gamma(p, e) = 1$, a contradiction.

In the second case $\eta_o(p, e) = 1$. But then, then there exists an object $q \in C(p, e)$ that is opposite to p and every object $q' \in C(p, e)\backslash\{q\}$ is assigned to direction $d(p, e)$ if p is assigned to direction $d(p, e)$. But then, if $\gamma(p) = 1 - d(p, e)$, then $f_\gamma(p, e) = 1$, a contradiction or otherwise if $\gamma(p) = d(p, e)$ and since every satisfying truth assignment of φ_0 is also a satisfying truth assignment of ψ, there exists exactly one object, q, in $C(p, e)$ that is assigned direction $1 - d(p, e)$ by the truth assignment corresponding to γ and thus, $f_\gamma(p, e) = 1$, a contradiction. □

As the next step we construct a 2-SAT formula φ_1 for all objects in H_1.

Construction 6.32. Let $\Phi := (x, y)$ be the guess of an instance $\mathcal{I} := ((N, X, \succ, C_n, \sigma_0, \sigma), e')$ of First Swap Reachable Assignment on cycle C_n. We will now construct a 2-SAT formula f_1 with parameter (p, e) as follows. Since $(p, e) \in H_1$ either $\eta_{1-d}(p, e) > 0$ or $\eta_d(p, e) = |C(p, e)|$.

If $\eta_{1-d}(p, e) > 1$ or $\eta_d(p, e) = |C(p, e)|$, then add to $f_1(p, e)$ the term

$$\begin{cases} \neg p & \textit{if } d(p, e) = 1 \\ p & \textit{otherwise.} \end{cases} \tag{6.16}$$

We further define φ_1 as follows:

$$\varphi_1 := \Big(\bigwedge_{(p,e) \in H_1} f_1(p, e)\Big) \wedge f_U \wedge f_D.$$

We will now show that the construction of φ_1 is correct.

Lemma 6.33. *Let $\Phi := (x, y)$ be a guess and let $(p, e) \in H_1$. Suppose there exists a valid selection for guess Φ. Let γ be a selection that respects Φ. If γ is valid, then it corresponds to a satisfying truth assignment of φ_1 and if γ corresponds to a satisfying truth assignment of φ_1, then it holds that $f_\gamma(p, e) = 1$ for all $(p, e) \in H_1$.*

Proof. We will show both directions of this statement by contradiction.

First suppose that γ is valid but it does not correspond to a satisfying truth assignment of φ_1. Then there exists an object-edge-pair $(p', e') \in H_1$ and a clause c in $f_1(p', e')$ such that c is not fulfilled by a truth assignment corresponding to γ. Then c is a clause as defined in Equation 6.16.

We distinguish between two cases. In the first case it holds that $\eta_d(p', e') = |C(p', e')|$, but then, according to Lemma 6.27, there exists no valid selection γ' and an object $q \in C(p', e')$ such that $\gamma'(q) = 1 - d(p', e')$. Thus, $f_\gamma(p', e') = 0$ and γ is not valid, a contradiction.

In the second case, it holds that $\eta_{1-d}(p, e) > 1$. But then there exist two decided objects q_0 and q_1 in direction $1 - d(p', e')$. Due to Lemma 6.27, there exists no valid selection γ' such that $\gamma'(q_0) \neq 1 - d(p', e')$ and $\gamma'(q_1) \neq 1 - d(p', e')$ and thus $f_\gamma(p', e') > 1$. Therefore, γ is not valid, a contradiction.

We will now prove the other direction of the statement. Suppose that γ corresponds to a satisfying truth assignment of $f_1(p, e)$ but $f_\gamma(p, e) \neq 1$. Since $(p, e) \in H_1$, $\eta_{1-d}(p, e) > 0$

or $\eta_d(p,e) = |C(p,e)|$. However, if $\eta_d(p,e) = |C(p,e)|$, then either $\gamma(p) = d(p,e)$ but then there exists no object $q \in C(p,e)$ such that p can be swapped with q at e and thus, γ is incomplete, a contradiction. Otherwise $\gamma(p) = 1 - d(p,e)$, but then according to the definition of f_γ, it holds that $f_\gamma(p,e) = 1$, a contradiction.

Now suppose that $\eta_{1-d}(p,e) > 0$ and then, due to Equation 6.16, $\gamma(p) = 1 - d(p,e)$ if γ corresponds to a satisfying truth assignment of φ_1. Thus, by definition $f_\gamma(p,e) = 1$, a contradiction. □

As the next step we construct a 2-SAT formula φ_2 for all objects in H_2.

Construction 6.34. Let $\Phi := (x,y)$ be the guess of an instance $\mathcal{I} := ((N, X, \succ, C_n, \sigma_0, \sigma), e')$ of First Swap Reachable Assignment on cycle C_n. We will construct a 2-SAT formula f_2 with parameter $(p,e) \in H_2$ as follows.

Every object in $C(p,e)$ is undecided and not opposite to p. First we compute $S(p,e,q)$ for every $q \in C(p,e)$. Let Q be the set of undecided objects q for which $S(p,e,q)$ is successful and let Q' be the set of undecided objects q for which $S(p,e,q)$ is unsuccessful.

If $|Q| = 0$, then we add the following term to $f_2(p,e)$.

$$\begin{cases} \neg p & \text{if } d(p,e) = 1 \\ p & \text{otherwise.} \end{cases} \tag{6.17}$$

If $Q = \{q\}$, then we add the following term to $f_2(p,e)$.

$$\begin{cases} p \to \neg q & \text{if } d(p,e) = 1 \\ \neg p \to q & \text{otherwise.} \end{cases} \tag{6.18}$$

Further we add the following term to $f_2(p,e)$.

$$\begin{cases} p \to \bigwedge_{q \in Q'} (q) & \text{if } d(p,e) = 1 \\ \neg p \to \bigwedge_{q \in Q'} (\neg q) & \text{otherwise.} \end{cases} \tag{6.19}$$

Lastly, we define φ_2 as follows:

$$\varphi_2 := \Big(\bigwedge_{(p,e) \in H_2} f_2(p,e) \Big) \wedge f_U \wedge f_D \wedge \psi.$$

We will now prove that the construction of φ_2 is correct. First we will prove this intermediate lemma that assists in proving the correctness of φ_2.

Lemma 6.35. *Let $(p,e) \in H_2$ be an object-edge-pair and let Q be the set of undecided objects $q \in C(p,e)$ for which $S(p,e,q)$ is successful. Then Q can be partitioned into two sets Q_0 and Q_1 and the following statement holds. There exists an $i \in \{0,1\}$ such that for every object $q \in Q_i$ the following clause exists in ψ.*

$$\begin{cases} q & \text{if } d(p,e) = 1 \\ \neg q & \text{otherwise.} \end{cases} \tag{6.20}$$

Proof. Recall that ψ is a 2-SAT formula such that every satisfying truth assignment of ψ is a harmonic selection.

We will first demonstrate how we partition Q into Q_0 and Q_1. Recall the definition of $S(p, e, q)$ which is the tuple (f, c), where f is the edge where q meets p and where $c = 1$, if q is closer to p in direction $d(p, e)$ than the guessed object $\Phi_{d(p,e)}$ and $c = 0$ otherwise. Since $S(p, e, q)$ is successful for all $q \in Q$, the edge where q meets p is equal amongst all $q \in Q$. Therefore we partition Q into two sets Q_0 and Q_1 as follows. Let $q \in Q$ and let $(f, c) = S(p, e, q)$. Then q belongs to the partition Q_c.

Now recall that since $(p, e) \in H_2$, p is undecided and thus, by definition of undecided objects, in a candidate list of $\Phi_{1-d(p,e)}$. Let e^* be the edge where p and $\Phi_{1-d(p,e)}$ can be swapped and which is on the same shared path of p and $\Phi_{1-d(p,e)}$ than e. Note that this shared path exists either before the first swap of p and $\Phi_{1-d(p,e)}$ or after the first swap of p and $\Phi_{1-d(p,e)}$.

We distinguish between two cases. In the first case, e^* is closer to p in direction $d(p, e)$ than e. But since every object q_1 in Q_1 is closer to p in direction $d(p, e)$ than $\Phi_{1-d(p,e)}$, according to Lemma 5.14, $\Phi_{1-d(p,e)}$ is shielded by q_1 in direction $1 - d(p, e)$. Thus, ψ contains the clause in Equation 6.20 for q_1.

In the second case, e^* is further away from p in direction $d(p, e)$ than e. But since every object q_0 in Q_0 is further away from p in direction $d(p, e)$ than $\Phi_{1-d(p,e)}$, according to Lemma 5.14, $\Phi_{1-d(p,e)}$ shields q_0 in direction $1 - d(p, e)$. Thus, ψ contains the clause in Equation 6.20 for q_0. $\square$

We can now prove that the construction of φ_2 is correct.

Lemma 6.36. *Let $\Phi := (x, y)$ be the guess of an instance $\mathcal{I} := ((N, X, \succ, C_n, \sigma_0, \sigma), e')$ of* First Swap Reachable Assignment *on cycle C_n. Suppose there exists a valid selection for guess Φ. Let γ be a selection that respects $\mathcal{I}$. For every $(p, e) \in H_2$ it holds that, if γ is valid, then it corresponds to a satisfying truth assignment of φ_2 and if γ corresponds to a satisfying truth assignment of φ_2, then $f_\gamma(p, e) = 1$.*

Proof. We will prove both directions of the statement by contradiction. First suppose that γ is valid but γ does not correspond to a truth assignment of φ_2. Then, there exists at least one clause c in φ_2 that is not fulfilled in γ. We distinguish between three different types of clauses.

Suppose c is a clause as in Equation 6.17. Then it holds that

$$\gamma(p) = d(p, e).$$

We distinguish between two cases. In the first case, there exists an object $q \in C(p, e)$ such that $\gamma(q) \neq \gamma(p)$. However, since in the case of Equation 6.17, there exists no undecided object q' in $C(p, e)$ such that $S(p, e, q')$ is successful and since $q \in C(p, e)$, also $S(p, e, q)$ cannot be successful. But then, according to Lemma 6.21, γ is not valid, a contradiction. In the second case there exists no object $q \in C(p, e)$ such that $\gamma(p) \neq \gamma(q)$. Thus, $f_\gamma(p, e) = 0$, a contradiction.

Now suppose that c is a clause as in Equation 6.18. Then there exists exactly one $q \in C(p,e)$ such that $S_{d(p,e)}(p,e)$ is successful but it holds that

$$\gamma(p) = d(p,e) = \gamma(q).$$

This means that one of the following two things is true. Either there exists an object $q' \in C(p,e)$ such that $S_{d(p,e)}(p,e)$ is not successful and it holds that

$$\gamma(p) = d(p,e) = 1 - \gamma(q'). \tag{6.21}$$

In the first case, due to Lemma 6.21, γ is not valid, a contradiction. In the second case, there exists no q' such that Equation 6.21 holds in which case $f_\gamma(p,e) = 0$, a contradiction.

Now suppose that c is a clause as in Equation 6.19. Then there exists an undecided object q such that $S(p,e,q)$ is unsuccessful but it holds that

$$\gamma(p) = d(p,e) = 1 - \gamma(q).$$

However, according to Lemma 6.21, then γ is not valid, a contradiction.

We will now prove the other direction of the statement by contradiction. Suppose that there exists a truth assignment γ of φ_2, but $f_\gamma(p,e) \neq 1$. We will first show that by construction of H_2, there exists at least one undecided object $q \in C(p,e)$ such that p and q are not opposite. Afterwards, we will make a case distinction to lead the above statement into a contradiction and thereby complete the proof.

By construction, H_2 excludes the set H_0 and hence all object-edge-pairs (p_0, e_0) such that $C(p_0, e_0)$ contains objects that are opposite to p_0. Further H_2 excludes the set H_1 and hence all object-edge-pairs (p_0, e_0) for which $\eta_d(p,e) = |C(p_0, e_0)|$ or $\eta_{1-d}(p,e) > 0$. Thus, for (p,e) there must exist at least one undecided object $q \in C(p,e)$ such that p and q are not opposite.

Let therefore $Q \subseteq C(p,e)$ be the set of undecided objects that are not opposite to p and such that $S_{d(p,e)}(p,e)$ is successful. We will now distinguish between the three cases where $|Q| = 0, |Q| = 1$ and $|Q| > 1$. In the first case $|Q| = 0$. But then according to Equation 6.17, for every truth assignment γ of φ_2 it holds that

$$\gamma(p) = 1 - d(p,e).$$

Then however, by definition, $f_\gamma(p,e) = 1$, a contradiction.

In the second case $|Q| = 1$. But then according to Equation 6.18, for every selection γ corresponding to a truth assignment of φ there exists an object $q \in C(p,e)$ such that it holds that

$$\gamma(p) = d(p,e) = 1 - \gamma(q).$$

Moreover, every object q' in $C(p,e) \backslash \{q\}$ is either decided in direction $d(p,e)$ and there exists a term in f_D that ensures that $\gamma(q') = d(p,e)$ or it is not decided but then $S(p,e,q')$ is not successful. However then, according to Equation 6.19, $\gamma(q') = d(p,e)$ if

$\gamma(p) = d(p,e)$. Thus, only q is assigned direction $1 - d(p,e)$ and hence $f_\gamma(p,e) = 1$, a contradiction.

In the third case it holds that $|Q| > 1$. Let therefore $q_0, q_1 \in Q$ such that $q_0 \neq q_1$. We distinguish between two cases. In the first case q_0 is closer to p in direction $d(p,e)$ than $\Phi_{d(p,e)}$ but q_1 is further away from p in direction $d(p,e)$ than $\Phi_{d(p,e)}$. Objects q_0 and q_1 are in this case of course interchangeable. Then, according to Lemma 6.35, q_1 is assigned to direction $d(p,e)$ by every selection corresponding to a satisfying truth assignment of ψ and, since φ_2 includes all clauses of ψ, q_1 is assigned to direction $d(p,e)$ by every selection corresponding to a satisfying truth assignment of φ_2. In the second case both q_0 and q_1 are both either closer to p in direction $d(p,e)$ than $\Phi_{d(p,e)}$ or further away from p in direction $d(p,e)$ than $\Phi_{d(p,e)}$. But then $S(p,e,q_0) = S(p,e,q_1)$ and, according to Lemma 6.20, q_0 and q_1 are in the same candidate list of $\Phi_{d(p,e)}$. From this it follows that there exists a clause $q_0 \neq q_1$ in f_U and thus, exactly one of q_0 and q_1 are assigned to direction $1 - d(p,e)$, regardless of the assigned direction of p.

Since one of the two cases is true for every $q_0, q_1 \in Q$, there exists exactly one $q \in Q$ such that $d(p,e) = 1 - \gamma(q)$. Thus, by definition, $f_\gamma(p,e) = 1$, a contradiction. From this our statement follows. □

After we proposed a 2-SAT formula for every object in the sets $H(\mathcal{C}), H_0, H_1$ and H_2 we will conjunct these formulas to construct the formula ϕ such that every satisfying truth assignment of ϕ is a valid selection and if a selection is valid, then it corresponds to a satisfying truth assignment of ϕ. After we showed this in the following proposition we will present the final theorem that shows how to solve Reachable Assignment on cycles in polynomial time.

Proposition 6.37. *Let $\Phi := (x,y)$ be the guess of an instance $\mathcal{I} := ((N, X, \succ, C_n, \sigma_0, \sigma), e')$ of* First Swap Reachable Assignment *on cycle C_n. Suppose that there exists a valid selection with guess Φ. Let γ be a selection that respects $\mathcal{I}$. Let*

$$\phi := (\bigwedge_{0 \leqslant i \leqslant 2} \varphi_i) \wedge f_U \wedge f_D \wedge \psi.$$

Then γ is valid, if and only if it corresponds to a satisfying truth assignment of ϕ.

Proof. Since $H(\mathcal{C}), H_0, H_1$ and H_2 are a partition of H, due to Observation 6.24, it holds that every object-edge-pair $(p,e) \in H$ is either in $H(\mathcal{C}), H_0, H_1$ or H_2. Let $(p,e) \in H$. We distinguish between four cases.

In the first case, $(p,e) \in H(\mathcal{C})$. Moreover, $H(\mathcal{C})$ is partitioned by $H(\mathcal{C}_D)$ and $H(\mathcal{C}_U)$. If $(p,e) \in H(\mathcal{C})_D$, then according to Lemma 6.27, if γ is valid, then it corresponds to a satisfying truth assignment of f_D and thus, by construction of ϕ also to a satisfying truth assignment of ϕ. Further, due to Lemma 6.27, if γ corresponds to a satisfying truth assignment of f_D, then $f_\gamma(p,e) = 1$. If $(p,e) \in H(\mathcal{C})_U$, then according to Lemma 6.29, if γ is valid, then it corresponds to a satisfying truth assignment of f_U and thus, by construction of ϕ also to a satisfying truth assignment of ϕ. Further, due to Lemma 6.29, if γ corresponds to a satisfying truth assignment of f_U, then $f_\gamma(p,e) = 1$.

In the second case, $(p, e) \in H_0$. If $(p, e) \in H_0$, then according to Lemma 6.31, if γ is valid, then it corresponds to a satisfying truth assignment of φ_0 and thus, by construction of ϕ also to a satisfying truth assignment of ϕ. Further, due to Lemma 6.31, if γ corresponds to a satisfying truth assignment of φ_0, then $f_\gamma(p, e) = 1$.

In the third case, $(p, e) \in H_1$. If $(p, e) \in H_1$, then according to Lemma 6.33, if γ is valid, then it corresponds to a satisfying truth assignment of φ_1 and thus, by construction of ϕ also a truth assignment of ϕ. Further, due to Lemma 6.33, if γ corresponds to a satisfying truth assignment of φ_1, then $f_\gamma(p, e) = 1$.

In the third case, $(p, e) \in H_2$. If $(p, e) \in H_2$, then according to Lemma 6.36, if γ is valid, then it corresponds to a satisfying truth assignment of φ_2 and thus, by construction of ϕ also to a satisfying truth assignment of ϕ. Further, due to Lemma 6.36, if γ corresponds to a satisfying truth assignment of φ_2, then $f_\gamma(p, e) = 1$.

Conclusively, if γ is valid, then it corresponds to a satisfying truth assignment of ϕ and if γ corresponds to a satisfying truth assignment of ϕ, then it holds that $f_\gamma(p, e) = 1$, for all $(p, e) \in H$. Thus, γ is sound. Further, γ is also a satisfying truth assignment of ψ and thus, as proven in Lemma 6.2, γ is harmonic. By definition, γ is then valid. □

We will now conclude this chapter by proving that Reachable Assignment on cycles is decidable in polynomial time.

Theorem 6.38. Reachable Assignment *on cycles is decidable in* $\mathcal{O}(n^3)$ *time.*

Proof. Let $\mathcal{I} := (N, X, \succ, C_n, \sigma_0, \sigma)$ be an instance of Reachable Assignment on cycle C_n. Given this instance we compute the 2-SAT formula ψ as described in Section 6.1. Constructing ψ requires to evaluate every pair of objects p and q whether p and q are compatible and whether p shields q from its destination or whether q shields p from its destination and thus, ψ can be constructed in $\mathcal{O}(n^2)$.

Afterwards we divide instance $\mathcal{I}$ into instances $(\mathcal{I}, e)$ of First Swap Reachable Assignment for each $e \in E(C_n)$, where e denotes the edge along which the first swap happens. This further determines the two objects x and y that are involved in the first swap, the guess $\Phi := (x, y)$ of the instance of First Swap Reachable Assignment.

For every such instance we proceed as follows. If there exists a candidate list $C(p, e)$ of a guessed object $p \in \{x, y\}$ such that every objects $q \in C(p, e)$ is decided in direction $d(p, e)$, then there exists no sound selection for this guess because p cannot be swapped with any object at edge e and thus, the guess is wrong. Otherwise we construct ϕ. If there exists a satisfying truth assignment of ϕ, then it is valid, otherwise there exists no valid selection as shown in Proposition 6.37. We distinguish between the two cases. In the first case there exists a satisfying truth assignment of ϕ. Then, that truth assignment corresponds to a valid selection and due to Proposition 5.15, there exists a swap sequence that transforms σ_0 in to σ for instance $\mathcal{I}$.

In the second case there exists no satisfying truth assignment of ϕ. Then, by Proposition 6.37, there exists no valid selection for guess Φ.

Thus, either we find a solution for one of the n instances or there exists no solution. Thus, we decide Reachable Assignment on cycles.

We will now conclusively prove that the above algorithm has a time-complexity of $\mathcal{O}(n^2)$. Constructing ϕ, we iterate over the set of objects and for every object p of those objects, in worst-case, add as many clauses to the respective sub-formulas of ϕ as there are objects in the candidate lists of p. Recalling Observation 6.25, this number is bounded by $2|X|$ where $|X|$ is the number of objects. Thus, constructing ϕ takes at most $\mathcal{O}(n^2)$-time. From this also follows that the number of clauses in ϕ is smaller than cn^2 where $c \in \mathbb{N}$ is some constant. Thus, ϕ can be constructed and solved in $\mathcal{O}(n^2)$-time. Since we create at most n formulas ϕ for each of the n possible guesses, the overall time-complexity of our algorithm is $\mathcal{O}(n^3)$. □

7 NP-hardness of Reachable Assignment on Cliques

In this chapter we will prove that Reachable Assignment is NP-hard on cliques. To do so, we will adapt the reduction of Reachable Object of general graphs to Reachable Assignment on general graphs by Gourves et al. [GLW17].

Theorem 7.1. Reachable Assignment *is NP-hard on cliques.*

Proof. Gourves et al. [GLW17] already propose a reduction of Reachable Object of general graphs to Reachable Assignment on general graphs. We denote the instance $\mathcal{I}$ of Reachable Object as the tuple $(N, X, \succ, G, \sigma_0, i, x_l)$ with agent i as well as reachable object x_l.

The strategy of Gourves et. al. is to create a copy of each agent in the original graph and connect each agent with its copy. This creates instance $\mathcal{I}' := (N \cup N', X \cup X', \succ'$ $, G', \sigma_0', \sigma')$. Sets N' and X' are copies of N and X respectively. Furthermore, the agents in N' are arranged such that $G'[N'] = K_{|N'|}$, a clique with $|N|$ vertices.

For each agent in the original graph it holds that they prefer the object the most that their copy initially holds. Furthermore, they do not contain any object that is initially held by another copy in their preference lists and the copies, except for the copy i' of i accept every object but x_l. However, i' prefers x_l the most.

If x_l then moves towards i, it cannot pass any agent in N' because the only agent that would accept x_l is i' but only i itself is connected to i'. Once x_l has reached i every agent in N swaps its current object with its copy in N', receiving its most preferred object. Note that now only i' holds its most preferred object among the copies.

These however are organized by Gourves et al. [GLW17] in such a way that they can get their most preferred object from some other copy.

We adapt this reduction to prove NP-hardness for Reachable Assignment on cliques. Bentert et al. [Ben+19] already proved NP-hardness for Reachable Object on cliques . To this end, we add all of the missing edges between the original vertices and the copies.

We will show that x_l is reachable for i in G if and only if every agent in G' can reach its most preferred object.

Suppose x_l is reachable for i. Then, the object x_l must have only visited agents from G so far. To show this suppose it was not true and there exists an agent from G, $u \in N$ such that $u \neq i$, that held x_l and swapped it with any copy. We constructed the preference lists of the copies in such a way that x_l only appears on the preference list of the copy of i. Further the preference list of the copy of i only contains the object x_l and the object that i prefers the most. However, then the only agent that can swap x_l with the copy of i is i itself. Thus, there cannot exist such an agent $u \neq i$, a contradiction.

We now showed that if x_l is reachable for i, then the object x_l must have only visited agents from G so far. That means that the copy of u in N', u', still holds the most preferred object of u. The preference lists are organized in such a way that u only accepts one copied object, the object initially held by u'. The object u' however accepts every object but x_l. Hence there is no way for u to swap x_l to any of the copies, contradiction. Since x_l is reachable for i but has so far visited only agents from G we proceed by swapping x_l, held by i with the object initially held by i'. The object x_l is the most preferred object of i' and i''s object the most preferred object of i. Since the assignment amongst the other copies has not changed we can now let each other agent from G swap their with their copy, hence receiving their most preferred object. Note that by now every agent in G possesses their most preferred object.

As in the original reduction by Gourves et al. [GLW17] the preference lists of the copies are constructed such that each of the copies can receive their most preferred object from another copy. Thus, after that procedure, every agent in G' reached their most preferred object.

Now suppose every agent is in possession of its most preferred object after some sequence of swaps. Then we know also that i' has received x_l, since that is i''s most preferred object. Since the only agent that would accept the object originally held by i' is i we know that i must have been swapping x_l to i' and hence was in possession of it at one point. Furthermore, as mentioned before there is no way for x_l to reach any of the copies unless i' but then only in exchange with agent i. Hence, x_l was reachable for i in G. □

8 Conclusion

In this work, we have investigated a generalization of the house marketing problem, a problem that stems from the field of *Multi-Agent-Systems*. This field studies, amongst other things, the efficient and fair distribution of resources among agents.

The generalization we studied here is called Reachable Assignment and was proposed by Gourves et al. [GLW17]. It is known to be NP-hard on general graphs. We have studied the problem of Reachable Assignment on two classes of graphs, namely cycles and cliques. For cycles we showed that there exists a $\mathcal{O}(n^3)$-time algorithm and for cliques we showed NP-hardness.

An open question remaining with regard to the Reachable Assignment problem is whether Reachable Assignment can be solved efficiently for several other graph classes. Recall that the key to solving Reachable Assignment on trees [GLW17] and on cycles was to exploit the number of unique paths an object can be swapped along. Finding graph classes in which this number is bounded and solving Reachable Assignment for these graph classes is a natural next step for further research. Moreover, since cycles are paths with one additional edge, it seems promising to investigate the parameterized complexity of Reachable Assignment with respect to the feedback edge number of the input graph next. Other possibilities are parameters of the preference profile as studied by Bentert et al. [Ben+19] for Reachable Object or to consider generalized settings such as allowing ties in the preference lists, as studied by Huang and Xiao [HX19] for Reachable Object.

Literature

[Ben+19] M. Bentert, J. Chen, V. Froese, and G. J. Woeginger. "Good Things Come to Those Who Swap Objects on Paths". In: *CoRR* abs/1905.04219 (2019).

[Che+06] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. A. Padget, S. Phelps, J. A. Rodríguez-Aguilar, and P. Sousa. "Issues in Multiagent Resource Allocation". In: *Informatica (Slovenia)* 30.1 (2006), pp. 3–31.

[Die12] R. Diestel. *Graph Theory, 4th Edition*. Vol. 173. Graduate texts in mathematics. Springer, 2012.

[GLW17] L. Gourvès, J. Lesca, and A. Wilczynski. "Object Allocation via Swaps along a Social Network". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI'17*. 2017, pp. 213–219.

[HX19] S. Huang and M. Xiao. "Object Reachability via Swaps along a Line". In: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI'19*. AAAI Press, 2019, pp. 2037–2044.

[Sha74] L. Shapley. "On Cores and Indivisibility". In: *Journal of Mathematical Economics* (1974), pp. 23–37.

[SW18] A. Saffidine and A. Wilczynski. "Constrained Swap Dynamics over a Social Network in Distributed Resource Reallocation". In: *Proceedings of the Algorithmic Game Theory - 11th International Symposium, SAGT'18*. Vol. 11059. Lecture Notes in Computer Science. Springer, 2018, pp. 213–225.