Optimisation of the waveguide routing for a telecommunication satellite

Marvin Stanczak

ONERA
THE FRENCH AEROSPACE LAB

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut Supérieur de l'Aéronautique et de l'Espace

Discipline ou spécialité :
Domaine mathématiques – Recherche opérationnelle

Présentée et soutenue par
Marvin STANCZAK
le : 11 mars 2022

Titre :
Optimisation du routage de guides d'ondes dans un satellite de télécommunications

---

École doctorale :
ED 309 – École Doctoral Systèmes (EDSYS)

Unité de recherche :
Office national d'études et de recherches aérospatiales (ONERA)

Directeurs de thèse :
M. Cédric PRALET ONERA
M. Vincent VIDAL ONERA

Encadrant industriel :
M. Vincent BAUDOUI Airbus Defence and Space

Rapporteurs :
M. Dominique FEILLET Professeur à l'École des Mines de Saint-Etienne
M. Nicolas JOZEFOWIEZ Professeur à l'Université de Lorraine

Autres membres du jury :
Mme. Nadia BRAUNER Professeure à l'Université de Grenoble Alpes
M. François CLAUTIAUX Professeur à l'Institut de Mathématiques de Bordeaux
Mme. Élise VAREILLES Professeure à l'Institut Supérieur de l'Aéronautique et de l'Espace

Résumé

Ces dernières décennies, la demande en services de communication fixes ou mobiles, de télévision en direct, de radio numérique ou d'Internet à haut débit a augmenté de manière exponentielle. Pour y répondre, les opérateurs de satellites de télécommunications doivent accroître continuellement la capacité de leurs satellites, ce qui engendre une hausse importante du nombre d'équipements et de connexions au sein des nouvelles charges utiles. Parmi ces connexions, les guides d'ondes sont des canalisations à section rectangulaire qui transportent des signaux électro-magnétiques entre deux composants du satellite. Ces signaux subissent des pertes radio-fréquentielles en ligne lors du parcours des guides d'ondes. Ainsi, la conception du harnais de guides d'ondes joue un rôle crucial sur les performances du satellite. Cette thèse propose des méthodes d'optimisation pour le routage détaillé des guides d'ondes permettant de réduire leur longueur tout en prenant en compte les contraintes de conception du harnais radio-fréquentiel.

Le problème de routage de guide d'ondes étudié, introduit dans la partie I, consiste à connecter une configuration d'entrée à une configuration de sortie en utilisant un guide d'ondes composé d'une succession de sections droites et de coudes (chapitre 1). Il possède plusieurs caractéristiques non standards pour les approches classiques de routage de canalisations (chapitre 2), tels que la gestion d'un ensemble de coudes restreint à un catalogue de coudes orthogonaux et/ou non-orthogonaux, ou bien la gestion d'une canalisation à section rectangulaire, ce qui rend important la notion d'orientation de cette canalisation.

Dans un premier temps, dans la partie II, toutes les contraintes d'espace de routage sont ignorées dans le problème de routage de guide d'ondes en espace libre (chapitre 4) et deux approches de résolution sont introduites. La première utilise la Programmation Linéaire Mixte (PLM) et est basée sur l'énumération des orientations possibles pour un segment du guide d'ondes (chapitre 5). Cette approche est néanmoins peu performante sur des instances issues de cas industriels, c'est pourquoi une autre formulation adaptée aux Algorithmes de Recherche Informés (ARIs) est proposée en utilisant une notion de plan de routage qui décrit un guide d'ondes partiellement routé (chapitre 6). La faisabilité d'un plan est évaluée avec de la programmation linéaire tandis que l'espace des plans est exploré avec des algorithmes comme le A* pondéré ou la recherche en faisceaux. Pour estimer la distance restante jusqu'à la destination, deux heuristiques utilisant la distance euclidienne et des combinaisons de coudes minimales sont présentées. Avec la meilleure heuristique, dont la consistance a été démontrée, cette seconde formulation surpasse l'approche PLM en résolvant la plupart des instances en moins d'une seconde (chapitre 7).

Dans la partie III, on s'intéresse au problème de routage de guide d'ondes en espace contraint, qui consiste à router un unique guide d'ondes dans un espace restreint pouvant contenir des obstacles. Pour modéliser ces contraintes spatiales, l'espace de routage est vu comme un espace continu en trois dimensions divisé en cellules convexes non régulières qui évitent les obstacles (chapitre 8). Les deux approches de résolution proposées précédemment pour le problème de routage de guide d'ondes en espace libre sont étendues à ce nouveau problème. Le choix d'un canal de cellules devant être traversées est introduit dans le modèle PLM (chapitre 9) et dans la formulation en problème de recherche. Par ailleurs, plusieurs heuristiques basées sur des pistes relaxées dans l'espace de routage sont proposées pour améliorer les estimations de distance à la destination en considérant les contraintes spatiales (chapitre 10). Finalement, si l'approche PLM testée n'est pas capable de fournir des solutions en un temps raisonnable, les ARIs résolvent certaines instances industrielles en proposant des guides d'ondes réalistes en quelques minutes (chapitre 11). Ces approches ont été intégrées dans des outils de conception industrielle des guides d'onde et ont permis de réduire le temps de conception du harnais radio-fréquentiel.

Abstract

In recent decades, the demand for fixed and mobile communication services as well as over-the-air television, digital broadcasting or broadband Internet has raised exponentially. To meet these growing needs, telecommunication satellite operators must continually increase the capacity of their satellites, which leads to a significantly higher number of components and connections inside the new satellite payloads. Among these connections, the waveguides are pipes with a rectangular section which carry useful electromagnetic signals between two components of the satellite payload. However, these signals suffer from on-line radio-frequency losses during their carriage along the waveguides. It results that the design of the waveguide harness plays a crucial role on the performances of the satellite. This PhD thesis proposes optimisation methods for the detailed routing of waveguides, reducing their lengths while taking into account the design constraints of the radio-frequency harness.

The studied Waveguide Routing Problem, introduced in Part I, consists in connecting an input configuration to an output configuration by using a waveguide composed of a succession of straight sections and bends (Chapter 1). It considers several non-standard features for classical Pipe Routing approaches (Chapter 2) such as dealing with a set of bends restricted to a catalogue that can contain both orthogonal and non-orthogonal bends, or with pipes of rectangular section, which makes the pipe orientation important.

As a first step, in Part II, all routing space constraints are ignored in the Free Waveguide Routing Problem (Chapter 4) and two resolution approaches are introduced. The first formulation uses Mixed Integer Linear Programming and is based on the enumeration of the possible orientations for the waveguide segments (Chapter 5). Because of the poor performances of this approach on industrial instances, another formulation adapted to the Informed Search Algorithms is proposed using a notion of routing plan that describes a partially routed waveguide (Chapter 6). The feasibility of a plan is then evaluated using Linear Programming while the space of plans can be explored with algorithms like Weighted A* or Beam Search. To do so, two different heuristics are proposed to estimate the distance to the destination using Euclidean distance and minimal bend combinations. With the best heuristic, which has been shown to be consistent, this second formulation clearly outperforms the MILP approach, solving most instances within a second (Chapter 7).

In a second phase, in Part III, the Constrained Waveguide Routing Problem, which consists in routing a single waveguide within a restricted three-dimensional space that may contain obstacles, is studied. To model these spatial constraints, the routing space is seen as a three dimensional continuous space divided into non-regular convex cells that avoid obstacles (Chapter 8). Then, both resolution methods introduced for the Free Waveguide Routing Problem are extended. The channel of cells to be traversed is first introduced as a set of new decision variables in the MILP model (Chapter 9) and in the Search Problem formulation. Furthermore, several heuristics based on relaxed trails in the routing space are proposed to improve the estimations by considering the space constraints and obstacles (Chapter 10). While the MILP approach tested is not able to provide solutions in a reasonable time, the Informed Search Algorithms solve small and medium industrial instances with realistic waveguides within a few minutes (Chapter 11). These approaches have been integrated into software tools for the industrial design of waveguides and have successfully reduced the time of design for the radio-frequency harness.

Contents

List of Figures

Part I

Introduction

Chapter 1

General Introduction

In recent decades, the demand for fixed and mobile communication services as well as over-the-air television, digital broadcasting or broadband Internet has raised exponentially. These services are now requested in more and more geographical areas, even in the least accessible ones, like oceans or deserts. To meet these growing needs, telecommunication satellite operators must continually increase the capacity of their satellites, that are able to reach areas inaccessible by terrestrial systems. This leads to a significantly higher number of components and connections inside the new satellite payloads (see Figure 1.1).

Figure 1.1 – Telecommunication satellites.

As a consequence, telecommunication satellite design phases become more complex and time consuming. To handle this complexity, optimisation methods are used to help engineers automating repetitive tasks, but also to improve the performances of the satellite and to approve its design. In particular, these optimisation methods are used all along the internal accommodation phase during which the components and the connections between them, including waveguides, are placed inside the payload. They allow designers to evaluate and optimise the routes of these connections, the goal being to reach the best performances and to reduce the manufacturing costs.

In this PhD thesis, the Waveguide Routing Problem (WRP) in a three-dimensional continuous space is studied. This problem has two main purposes. The first one consists in proposing a route for a waveguide between two components of the payload. The route should satisfy waveguide design rules as much as possible in order to minimise the modifications afterwards. The second one is to provide an estimation of the Radio-Frequency Losses (RF-losses) along the waveguide, which depend directly on the length of the route. This problem is clearly related to standard Pipe Routing (PR) problems presented in Chapter 2.

In the context of the Industrial Agreement of Training through Research or "Convention Industrielle de Formation par la REcherche" in French (CIFRE) and of the collaboration between Airbus Defence and Space (Airbus DS) and the French Aerospace Lab or "Office National d'Études et de Recherches Aérospatiales" in French (ONERA), the objective of this PhD thesis is to propose innovative solutions to deal with the Waveguide Routing Problem.

1.1 Industrial context

This section provides a basis for understanding the issues related to telecommunication satellite design.

1.1.1 Telecommunication satellites

A telecommunication satellite is made up of a platform and a payload. The *platform* ensures the station-keeping in orbit of the satellite as well as its thermal monitoring, its electric control, and the interface with the ground segments, that is to say the control centres. On the other hand, the *payload* fulfils the mission of the satellite which consists in receiving an electromagnetic signal from the Earth, amplifying it, and transmitting it to another geographical area. The payload is the main part of a telecommunication satellite.

A *broadband signal* is emitted from the Earth and travels about 36000 kilometres to reach a geostationary telecommunication satellite. The received signal, called *upstream signal* or *forward signal*, is very weak, about 10^{-10} watts, because of the attenuation along the path to reach the satellite. So, the signal should be strongly amplified, up to 10^2 watts, before sending it back to Earth in an exploitable way. This amplification is preceded by a frequency translation which avoids the interference phenomenon between the upstream signal and the transmitted signal, called *downstream signal* or *return signal*. Figure 1.2 shows the travel of the signal.

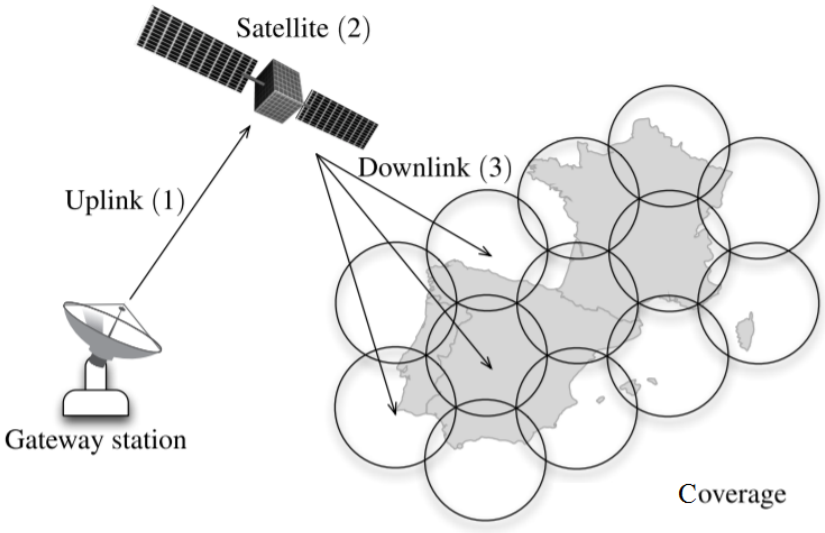


Figure 1.2 – Mission of a telecommunation satellite.

1.1.2 Communication payload

The payload is made up of a huge number of units which are fixed on the walls and panels forming the structure of the satellite (see Figure 1.3). Each one plays a clear role in the amplification chain. Most telecommunication satellites have a similar architecture depicted in Figure 1.4. The classical course of a broadband signal in a satellite can be described as follows.

A wide-band signal is composed of several narrow-band signals called *channels*. When the broadband signal is received by the reception antenna, it is amplified a first time with a minimal noise using a Low Noise Amplifier (LNA). The power of the signal is only a few hundred picowatts at the input of the LNA and is a few microwatts at its output. In order to minimise losses, LNAs are placed as close as possible to the reception antenna.

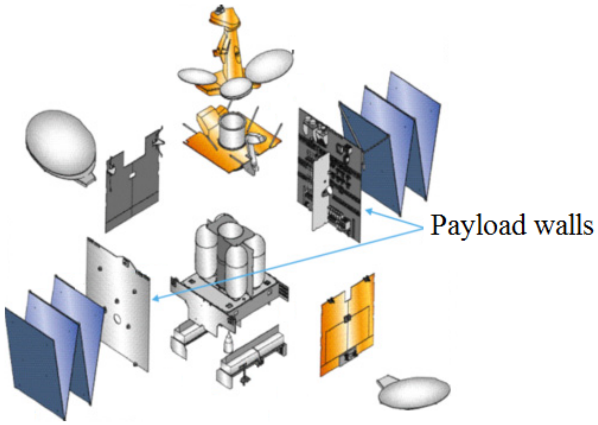


Figure 1.3 – Structure of a telecommunication satellite.

Then, a frequency down conversion is applied with a Down Converter (D/C). This conversion makes possible the differentiation between the upstream and downstream signals. In practice, a signal mixer couples the useful signal of frequency F_u with a signal of frequency F_{LO} to obtain a useful signal frequency translated with the frequency F_{LO}. For instance, the order of magnitude of the conversion is a few GHz in the C-band, which is a frequency band widely used in telecommunication.

The core of the amplification chain consists in amplifying the broadband signal using Travelling Wave Tube Amplifiers (TWTA) (see Figure 1.4). They are the main components of the payload and consume from 90% to 95% of the electrical power available on the satellite. They strongly amplify a channel using energy trade with an electron beam. However, a TWTA is not able to strongly amplify a broadband signal. This is why the broadband signal is split into channels using an Input Multiplexer (IMUX) before the amplification. The channels are then recombined using an Output Multiplexer (OMUX) after the TWTAs to form the downstream signal. So it is necessary to have at least a number of TWTAs equal to the number of channels in order to process all channels simultaneously.

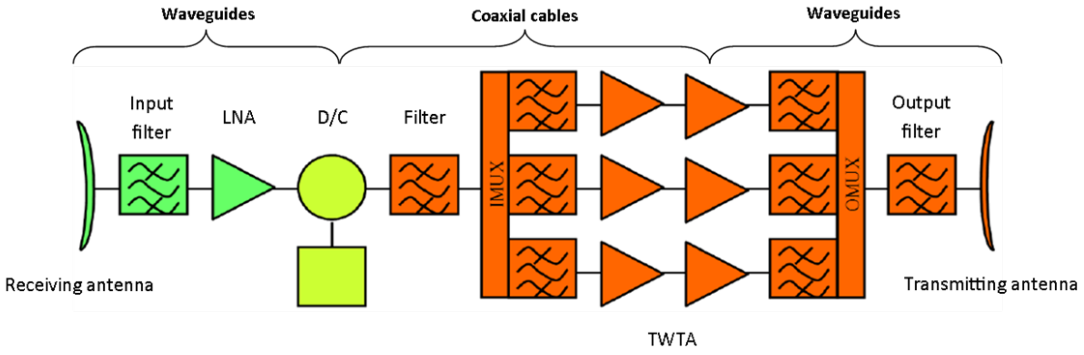


Figure 1.4 – Amplification chain.

Nevertheless, telecommunication satellites can suffer from component failures during their lifecycle. To address the breakdowns of TWTAs, designers set up a redundancy mechanism. Because of the high price of a TWTA (about tens of thousand euros), only one redundant TWTA is used for 5 or 6 nominal TWTAs. In the case of a failure, a channel can be redirected to a redundant TWTA by means of two symmetrical switch matrices, called *redundancy matrices*.

A *switch* is a unit that allows redirecting electromagnetic signals inside the payload by linking an input connector to an output one. Depending on the position of the switch, the signal from the input connector is propagated into one output or another. The first switch matrix located between the input channels and the tube amplifiers guarantees the access to any redundant TWTA if needed, or even to all TWTAs at the same time. The second switch matrix ensures the redirection of the signals getting out of the TWTAs to the output channels.

Last, a telecommunication satellite can generally cover several geographical areas, thanks to several antennas. Processing signals coming from several areas simultaneously requires other switch matrices, called *selectivity matrices*. These matrices, placed before the first redundancy matrix and after the second one, provide the possibility to select the channels to process, called *active channels*. These active channels define the mission of the satellite.

Thus, the whole architecture of a telecommunication satellite contains an important number of components traversed by the broadband signal. This implies a huge number of connectors to carry the signal between these components.

1.1.3 Radio-frequency harness

The set of connectors between the components of the amplification chain is called the Radio-Frequency Harness (RF-harness). Two kinds of connectors can be distinguished.

The first ones are the coaxial cables. A *coaxial cable* is a flexible connector which uses an internal conductor surrounded by insulating layers and a shielded outer layer to keep the signal-to-noise ratio as low as possible. It can carry digital or analog signals of low or high frequency.

The other kind of radio-frequency connectors are the waveguides. A *waveguide* is a pipe, that means a rigid and hollow connector, which carries an electromagnetic signal. Usually, the cross-section of a waveguide is rectangular and its size (given by its width a and its height b), also called *gauge*, depends on the frequency band of the signal (see Figure 1.5). Table 5 on page 167 in the appendices provides some examples of common gauges used in telecommunication satellites.

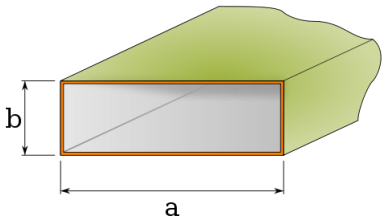


Figure 1.5 – Gauge of a waveguide.

Waveguides are more expensive than coaxial cables. Furthermore, they are more complex to design and place inside the payload because of their rigidity. However, the main advantages of the waveguides are their low attenuation of the signal and their strong heat resistance. For these reasons, waveguides are generally used before the first amplification, between the receiving antenna and the LNAs to prevent in-line attenuation of the weak upstream signals. They are also used after the TWTAs where the strongly amplified signals induce high temperatures. Thus, the coaxial cables are mainly used between the LNAs and the TWTAs as shown on Figure 1.4 on the preceding page.

1.1.4 Payload sizing

During the design of a telecommunication satellite, payload sizing plays a crucial role. Indeed, the number of components (in particular the number of TWTAs) and their position as well as the connectors between them define the physical and technical characteristics of the satellite, like its heat dissipation or its electric power storage capacities. The final design of the payload results from several iterations. Each iteration consists in approving a proposal with respect to technical, functional and financial specifications. If it is not compliant, the design is modified and revalidated until compliance.

In the case of a telecommunication satellite, the preponderant feature is the amplification capacity which should be guaranteed in spite of RF-losses encountered all along the amplification process. These losses appear through the dissipation as heat of the electric power inside the units. Dissipation occurs during the amplification with a fixed loss and during the carriage of the signal with the attenuation of the power along waveguides and coaxial cables. It depends almost linearly on the length of these connectors.

An accurate estimation of these losses is necessary to assert that the payload meets the amplification requirements, but, more importantly, their limitation is essential because significant losses result in the increase in the power or in the number of TWTAs to compensate the power loss. Such modifications require a greater heat dissipation and electric power storage capacities in order to face higher dissipation and electric power demand.

The RF-losses can be limited by minimising the lengths of the waveguides and coaxial cables, but also by optimising the layout of the components in the satellite to reduce the distance between them. Nevertheless, for an optimal efficiency, each component should be maintained in a given temperature range. This is a major issue during satellite accommodation, especially for the highly dissipative units like the TWTAs which dissipate about 60% of the electric power they consume. The components are placed according to their optimal temperature range into hot or cold areas created using heatpipes (heat conductors inside and on the panels of the satellite). Thus, the equipment layout is very constrained and the losses are mainly reduced through the minimisation of the coaxial cable and waveguide lengths in the RF-harness. This is achieved by the routing of the connectors.

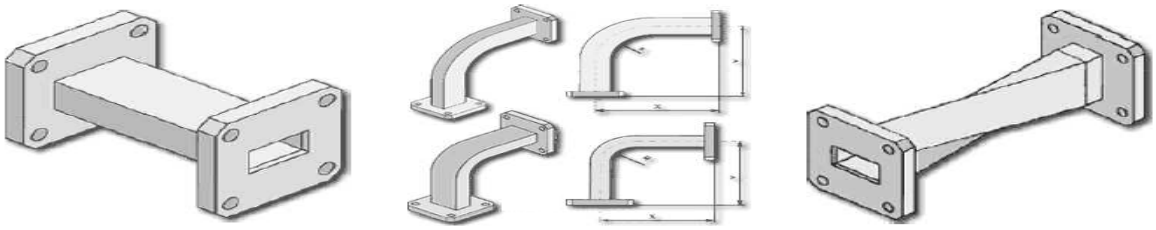
1.2 Waveguide Routing

This PhD thesis focuses on Waveguide Routing (WR) for the RF-harness of a telecommunication satellite. It consists in linking pairs of ports that should be connected using a waveguide. The design of the waveguides in the RF-harness follows three stages:

1. *Space Reservation*: a first design of the waveguides with non-optimised routes is proposed to guarantee the feasibility of the RF-harness and to preserve space for the waveguides in the future iterations with the other design teams;

2. *Detailed Routing*: the design of each waveguide is optimised by considering the best candidate route, reducing the bend radii and saving length and bends as much as possible;

3. *Iterative Routing*: the waveguides are rerouted during several iterations resulting from discussions with the other design teams like the mechanical stress team, the thermal team, or the electrical harness team.



(a) Straight section. (b) Bends. (c) Twist.

Figure 1.6 – Waveguide parts.

Furthermore, the routing of a waveguide between a pair of ports is also a three-step approach. First, the geometry of the route is created by routing a single waveguide between both ports. Then, this waveguide is split into several parts according to the maximal length for a waveguide (about 1 meter). A flange is placed at the ends of each section in order to assemble them into the complete geometry (see Figure 1.6 on the previous page). Last, the waveguides are fixed to the satellite walls using posts and brackets as shown on Figure 1.7. The latter must be regularly placed along each waveguide respecting a maximal distance of 300 millimetres between two brackets and trying to place them as close as possible to the flanges for stability reasons.

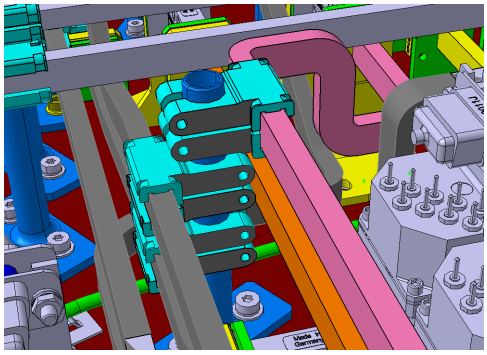


Figure 1.7 – A bracket grouping several waveguides.

Waveguide Routing is one of the most complex and time consuming phases of payload design. With the ever-increasing demand in capacity in telecommunication satellites, the number of waveguides has exponentially raised inside the payloads, reaching several thousands in the more recent platforms. Moreover, waveguide designers must deal with an important number of constraints, presented in Section 1.2.1, to guarantee the manufacturability and the operability of the RF-harness, but they must also minimise RF-losses (see Section 1.2.2). As a result, the manual Waveguide Routing methodology requires numerous iterations between the design and Computer-Aided Design (CAD) teams over a period of 4 to 6 months. An example of waveguide routing between several switch blocks on Figure 1.8 illustrates the complexity of the task.

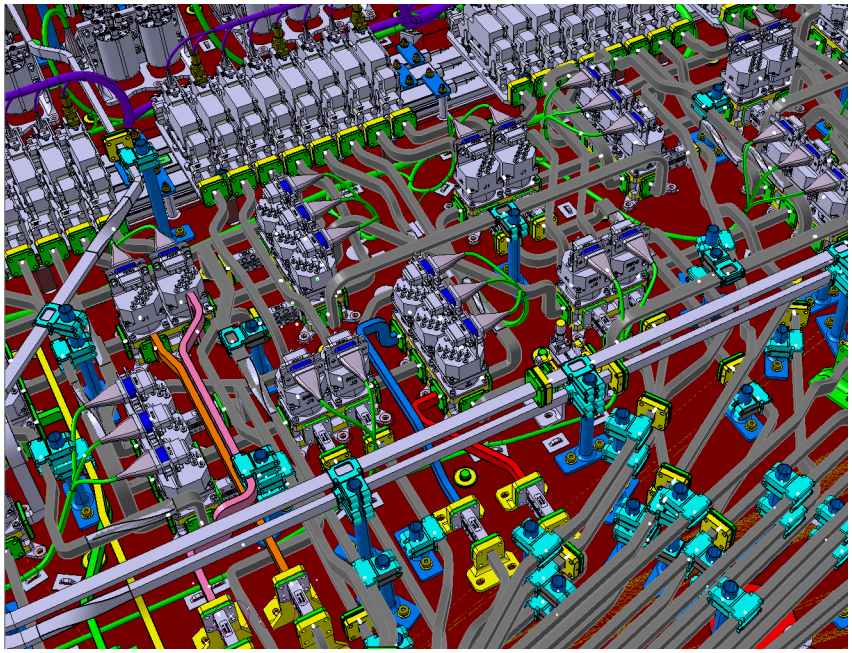


Figure 1.8 – Example of a hand-made waveguide routing.

1.2.1 Waveguide Routing rules

The route of a waveguide must satisfy many constraints which can be classified into five types that are detailed in this section: functional, manufacturing, conflict-avoidance, attachability and operability constraints.

Functional constraints

First of all, to fulfil its functional mission, a waveguide should connect an origin to a destination. Both of them are materialised by ports located on components. Ports can be located at different heights from the wall of the satellite, or even on different panels and can also have distinct orientations as shown on Figure 1.9. Thus, the starting and ending parts of the waveguide must have specific orientations, which are usually normal to the plane that contains the associated port. These constraints are called *functional constraints*.

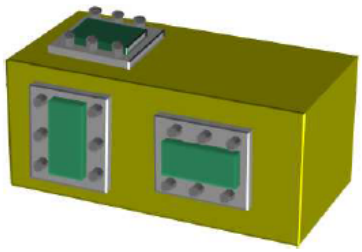


Figure 1.9 – A unit with 3 ports.

Manufacturing constraints

Manufacturing constraints guarantee that the waveguide can be produced by a manufacturer. Today, there are several methods to manufacture waveguides with *bends*, that means with orientation changes. Traditionally, manufacturers heat a long straight section with the length of the whole waveguide, then they bend it at the desired angles using pliers. Bends built this way, called *formed bends*, require that the *straight sections* of the waveguide have a length larger than the width of the pliers. For this reason, any straight section of a waveguide must have a minimal length (about 5 millimetres).

Another way to build waveguides is to manufacture bends separately using particular processes and to solder straight sections and bends together afterwards. This manufacturing method using *machine bends* allows generally smaller bend radii than the formed bends.

More recently, with the progress of 3D printing techniques, more complex waveguides can be built in one piece without any limitation on the straight section length. However, printed waveguides are not widely used today because of their high RF-losses. The poor quality of the signal propagation is mainly due to the roughness of the surface obtained by 3D printing.

Thus, the waveguides in the RF-harness use mainly formed and machine bends. The available ranges of angles and bend radii of these bends are provided by manufacturers and depend on the gauge. Nevertheless, a set of standard bends has been defined in a *catalogue* for each gauge in order to maximise the reuse of similar bends. This reduces RF-harness manufacturing costs by avoiding to place multiple orders for highly customised waveguides.

These catalogues contain orthogonal bends (90°) and non-orthogonal ones (45°, 30°, or 60°). They can also contain a special piece, called *twist*, which changes the orientation of the cross-section around the axis of the current segment, as shown on Figure 1.6c on page 7. Twists are more expansive than a 3-bend combination that performs the same operation, but they save space in some cases.

The last manufacturing constraint consists in avoiding some sequences of bends because the resulting waveguide cannot be produced by a manufacturer. In this case, the waveguide is said to be *infeasible*.

Conflict-avoidance constraints

A waveguide must also satisfy *conflict-avoidance constraints*. The most obvious one is that it must not conflict with an existing component or another waveguide of the payload. This is one of the hardest aspects of the waveguide designers' work, because a lot of units and waveguides are involved (see Figure 1.8).

Attachability constraints

Designers must also ensure that each segment of the waveguide can be fixed on the walls or panels. Indeed, brackets are regularly placed along the geometry to guarantee the robustness to mechanical stress, especially during the launch of the satellite. To do so, the orientation of a waveguide section is constrained depending on the panel it is routed on. In concrete terms, a waveguide segment must have at least one edge of its cross-section orthogonal to the normal of its fixing wall to be attachable, as illustrated on Figure 1.10.

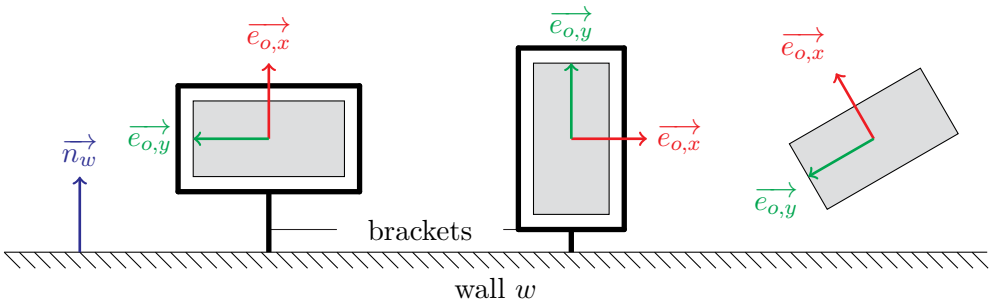


Figure 1.10 – From left to right, two attachable orientations and a non attachable one.

Moreover, the infeasible waveguides mentioned earlier systematically generate non attachability issues. Indeed, because of the sequence of bends they use, these waveguides always contain at least one non attachable segment whatever the fixing panels or the origin orientation. So, two kinds of attachability constraints can be distinguished:

- *global attachability constraints* that depend only on the sequence of bends used in the waveguide and are therefore intrinsic constraints;
- *wall-dependent attachability constraints* that depend on the walls on which the segments of the waveguide are routed. These constraints are said to be extrinsic.

Operability constraints

Last, the *operability constraints* ensure that the operators of the Assembly Integration and Test (AIT) team will be able to mount the waveguides of the RF-harness during the assembly phase. The main goal of these constraints is to guarantee the access to the mounting screws of the flanges and to preserve sufficient space for the assembly/disassembly operations.

For instance, a minimal distance must be respected between a flange and the first bend as illustrated on Figure 1.11. This minimal distance depends on the type of bend. Usually, bends along the widest side of the waveguide, called *H-plane bends*, do not block the access to the screws and a short distance is acceptable (about 15 millimetres). On the opposite, bends along the shortest side, called *E-plane bends*, require a bigger clearance distance before the first bend (about 30 millimetres) because they would obstruct the screws of the flange.

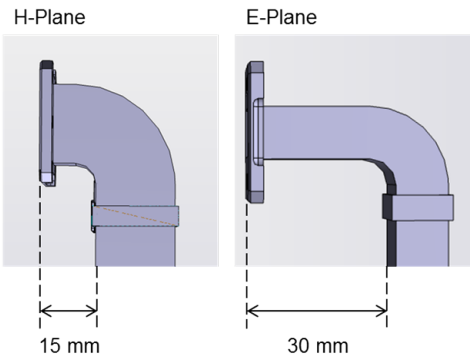


Figure 1.11 – Minimal distance between a flange and a bend.

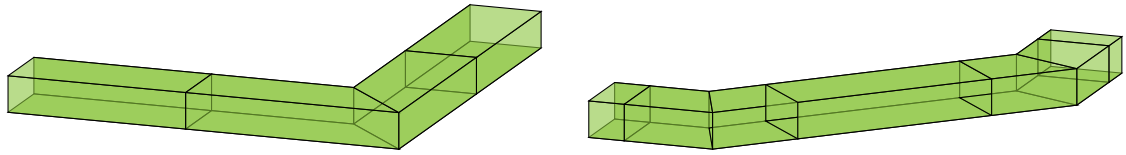
Another minimal distance must be respected between two waveguides (about 5 millimetres) for the same reasons.

The minimal length of the straight sections can also be seen as an operability constraint. It is sometimes necessary to ensure a minimal distance between two successive bends to satisfy the mounting needs. In this case, the minimal distance corresponds to 3 times the width of the gauge, that is $3a$.

1.2.2 Waveguide cost minimisation objective

As explained previously, the main objective in Waveguide Routing is to estimate and minimise the RF-losses of the RF-harness. The first contribution comes from in-line losses and depends linearly on the waveguide length which should be minimised. In addition, bends generate fixed losses according to their type, so the number of bends must be minimised too. In some cases, this second criterion is contradictory with the in-line criterion because a solution using more bends can be shorter than a solution with fewer bends (see Figure 1.12).

It can be noticed that the minimisation of the total length has other benefits. Indeed, reducing the total length makes the RF-harness lighter which saves weight in the satellite and has a positive impact on launching costs. In the same way, obtaining shorter waveguides that use fewer bends leads to a cheaper RF-harness in terms of manufacturing costs.



(a) 1-bend solution. (b) 2-bend solution with a shorter length.

Figure 1.12 – Trade-off between saving length and bends.

1.2.3 Existing waveguide routing tool

The Multiple Waveguide Routing Problem (MWRP) in a telecommunication satellite, which consists in routing several waveguides, has already been studied by Bessaih [9]. They addressed the problem by uniformly discretising the routable space into regular cells and by defining the route of a waveguide as a path in the adjacency graph of these cells. The proposed resolution approach sequentially routes waveguides using first a shortest path algorithm with a large neighbourhood, and then a repairing function that locally corrects conflicts between a waveguide and the already routed ones.

From this work, a waveguide routing tool has been implemented and integrated in the Airbus DS software suite in order to help engineers facing the increasing payload complexity (see Figure 1.13). The optimisation method provides very good estimations of the RF-losses and, coupled with a component positioning tool, significantly reduces the RF-harness design time, especially during the bid phase when a preliminary design must be proposed to prove the satisfaction of customer specifications.

However, the routes proposed by this algorithm are hardly ever used by designers as final routes. Indeed, the generated waveguides suffer from several defects which force engineers to rebuild them from scratch. The main reason is that several constraints are violated because of the discretisation. For instance:

- the starting and ending points of the proposed routes do not exactly match with the real positions of the ports, because the latter are not placed on a regular grid;
- the angles of the orientation changes along the routes are non-standard and can even correspond to infeasible waveguides;

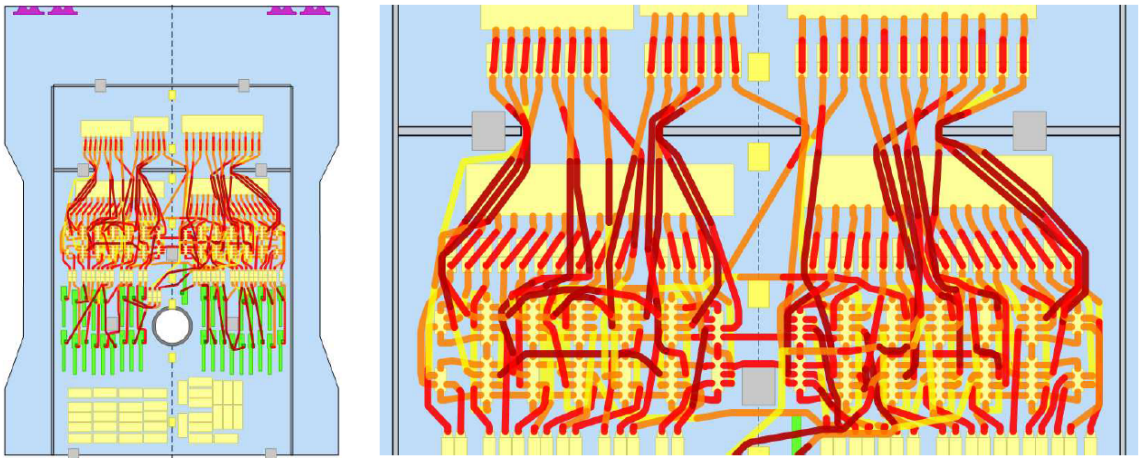


Figure 1.13 – Results of the existing tool on a wall.

- the bend radii are not taken into account, which leads some straight sections to violate the minimal length constraint after rebuilding the waveguide.

Besides the anomalies due to the discretisation, the approach gives solutions containing a lot of bends and does not consider the orientation of the cross-section along the route. For all these reasons, the existing tool provides unrealistic waveguides which are not used by designers, except to have a general idea of the waveguide paths as well as the order in which they must be routed.

1.3 Contributions

This PhD thesis proposes to deal with the raising complexity of communication payloads by defining a new optimisation method to route in a detailed manner a single waveguide of the RF-harness. The main goal is to automatically generate waveguides that are as realistic as possible in order to minimise modifications afterwards by waveguide designers. In this purpose, the main existing Pipe Routing techniques are presented in Chapter 2 and a state of the art about Search Algorithms (SAs), which are widely used to solve the Pipe Routing Problem, is detailed in Chapter 3. Then, the issue is addressed in two steps of increasing complexity.

First, the Free Waveguide Routing Problem (FWRP) is studied in Part II. It consists in routing a single waveguide in a three-dimensional continuous space free from any obstacle nor space constraint (see Chapter 4). The particularity of the proposed approaches is to be able to deal with non-orthogonal bends defined by a catalogue and with unsymmetrical cross-sections, the goal being to satisfy all the constraints of the Waveguide Routing Problem. Introduced algorithms are based on the enumeration of the possible orientations of the waveguide segments and on Linear Programming (LP). First, an exact method, which theoretically provides optimal solutions in a finite time, is considered in Chapter 5 using Mixed Integer Linear Programming (MILP). To challenge this exact method that can be time-consuming, a Search Problem (SP) formulation is introduced in Chapter 6 by defining the notion of routing plan to describe a partially routed waveguide. This formulation makes it possible to solve the FWRP using Informed Search Algorithms (ISAs), like A*, Weighted A* Search, Beam Search or Steeple Ascent Hill-Climbing for which a state of the art is presented in Chapter 3. They use heuristic and evaluation functions to explore the search space of routing plans. Two evaluation methods based on LP models, Euclidean distances and minimal bend combinations are proposed. Experiments conducted on instances inspired from an existing satellite and using realistic bend catalogues clearly show that ISAs can outperform the proposed MILP approach while providing comparable

guarantees on the solution quality. The most competitive ISA solves each test instance within a second on a standard laptop.

Secondly, in Part III, the Constrained Waveguide Routing Problem (CWRP), which extends the FWRP by constraining the waveguide to be contained in a routable space, is addressed. A methodology is presented in Chapter 8 to build the routable space from the satellite walls as an adjacency graph of polyhedral traversable cells that avoid obstacles. Then, the MILP and SP formulations are respectively adapted in Chapter 9 and Chapter 10 to take space constraints into account by considering the sequence of cells crossed by the waveguide. Since the heuristic functions proposed for the FWRP ignore space constraints, new heuristics are also introduced. They are based on a graph of relaxed routes, called *trails*, which connect each cell to the destination while staying inside the routable space. Thus, the remaining distance as well as the bend combination to use can be more accurately estimated using the shortest trail for a routing plan. Both MILP and SP formulations have been experimented on simple cases inspired from the industry. Again, the MILP method is outperformed by the ISAs. The solutions found are completely acceptable for waveguide designers even if there is no more guarantee on their quality. The Weighted A* Search algorithm, which is the fastest method, solves within several minutes simple cases that can be encountered in an industrial context.

Note that the appendices in Part V contain a description of the experimentation data and a glossary that lists all the notations used in this thesis.

Chapter 2

State of the Art about Pipe Routing

Waveguide Routing is a particular case of Pipe Routing which has been widely studied since the 70's. Pipe Routing is involved in various industrial contexts like plant layout [79, 85, 35], aeronautics [95, 96, 97], shipbuilding [45, 71, 5] or Very Large Scale Integration (VLSI), as shown on Figure 2.1. In this chapter, a *pipe* may refer to a rigid connection that carries gaz or liquid (water, fuel, ...), to a metal track between two components of an integrated circuit or to a waveguide. In most applications, Pipe Routing is used to evaluate the optimality of a layout, but it is a complex task that is not fully automated. It requires specialized engineers who use their experience to manage a large number of pipes at cost of a time-consuming design phase. A lot of research has been carried out to propose approaches that reduce human intervention during the routing phase.

(a) Aeronautics (b) Plant layout (c) VLSI

Figure 2.1 – Applications of Pipe Routing.

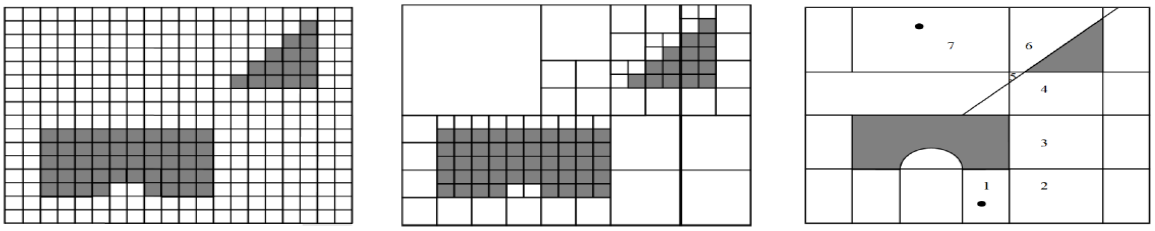
Pipe Routing consists in finding a set of routes for a set of pipes, where each pipe is defined by a set of terminals that must be interconnected while satisfying various constraints. A *terminal* is an area, generally a point, on a component which enables the connection with others components. Depending on the application, the set of terminals for a pipe may be composed by a source and a destination, or a source and several destinations. In the same way, the constraints that the routes must satisfy change according to the application. The objective is to minimise the pipe length and avoid physical conflicts with components or other pipes. Often, the number of direction changes along the routes must be minimised too for economical and quality reasons. For instance, when the pipe is carrying a liquid, bends generally deteriorate the flow rate. The quality of a piping system can also depend on the number of brackets which are needed to fix the pipes. In this case, the number of pipes per bracket must be maximised. Then, for safety reasons, pipes must generally satisfy a minimal distance from particular components. Maintenance and operability constraints may require providing access to mounting elements (like screws) and valves, or routing pipes along walls. This list of constraints is not exhaustive.

In this chapter, four existing approaches to solve the Pipe Routing Problem are presented: Cell Decomposition Approaches in Section 2.1, Line Search Algorithms in Section 2.3, Skeleton Approaches in Section 2.2 and Parametric Models in Section 2.4. Finally, Section 2.5 presents the motivations of this thesis with regards to the existing methods.

2.1 Cell Decomposition Approaches

2.1.1 Principle

The most widely used method to deal with the Pipe Routing Problem is the Cell Decomposition Approach (CDA). In this approach, the routing environment is discretised into a set of cells $\mathcal{C}$. In most cases, the decomposition is based on a regular grid and is homogeneous in the sense that all cells have identical dimensions, as illustrated on Figure 2.2a. So a cell is generally a rectangle in a 2D-space or a rectangular parallelepiped in a 3D-space. Nevertheless, heterogeneous decompositions where cells have different dimensions can be built using a quadtree for instance (see Figure 2.2b). These decompositions do not depend on obstacle shapes and cells which overlap with obstacles are labelled as occupied. To reduce the number of cells, it is possible to use obstacle shapes to create cells such that the union of all cells is exactly the available routing space, as shown on Figure 2.2c. However, such a decomposition is more complex and its computation may take a long time. By contrast, obstacle independent decompositions like regular grids generate simple cells, although the boundary of obstacles is approximated.



(a) Regular grid. (b) Quad tree. (c) Convex decomposition.

Figure 2.2 – Cell Decomposition Approaches.

2.1.2 Maze Routing Algorithms

From a decomposition of the routing space, a pipe can be routed as a path in the adjacency graph of cells $G(\mathcal{C}, \mathcal{I})$, where $(c, c') \in \mathcal{I}$ if cells $c \in \mathcal{C}$ and $c' \in \mathcal{C}$ have a common boundary, as shown on Figure 2.3. Routes are then computed using Maze Routing Algorithms (MRAs) [66, 61]. The most famous one is LEE's algorithm [59] which has been proposed to solve 2D routing problems in VLSI. Pipes are routed in a sequential way, one after the other, using a procedure similar to DIJKSTRA's algorithm [24] in the adjacency graph $G(\mathcal{C}, \mathcal{I})$. Starting from the source cell, the best route for a pipe is computed by exploring the undiscovered cells in the neighborhood of a frontier until the target is reached. The cost of a route is defined using a notion of mass associated with each cell and depending on the length of the path, the number of bends used, the number of crossings with existing pipes or the distance to obstacles or other pipes.

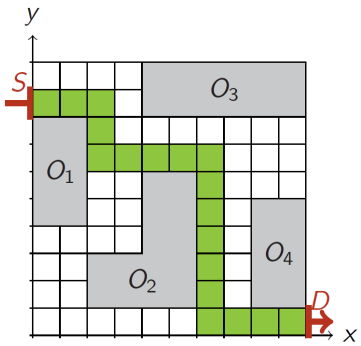


Figure 2.3 – A routed pipe as a path in an adjacency graph of cells.

2.1.3 Improved variants

The breadth-first search of initial MRAs can be replaced by more efficient search algorithms which consume less memory space and/or address pipe routing problems with more complex constraints. Five of them are described below.

Knowledge Based Engineering

In plant layout and aeronautics (see Figure 2.4), several approaches have been proposed to apply MRAs in combination with Knowledge Based Engineering (KBE) methods which consist in identifying and implementing rules and knowledge inspired by the manual routing process. For instance, in [96], the A* algorithm introduced by HART [37] is used as a routing procedure. It selects the most promising cell to explore using an evaluation function that estimates the distance to the target (see Section 3.2.1) but also considers specific constraints through penalisation costs, like bend radius and safety distance between pipes. Another possibility is to apply rules to define the order in which pipes are routed [50, 79]. Typically, wider pipes are routed before smaller ones.

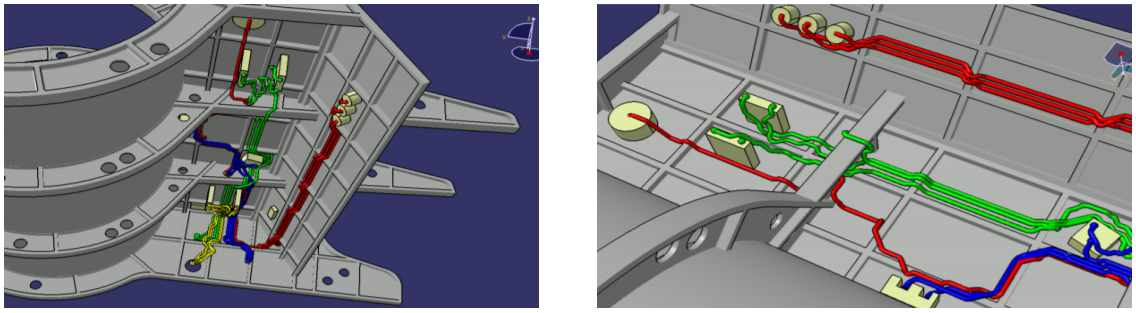


Figure 2.4 – Pipe routing in aeronautics.

Genetic Algorithms

ITO [44, 45, 46] uses a Genetic Algorithm (GA) in a routing space decomposed in homogeneous rectangular cells. A chromosome represents a candidate route and is defined as a vector of states with five possible values: a state that corresponds to having reached the destination and a state for each direction (right, left, up and down). The initial population is generated using several mechanisms that provide diversified routes. Preferential directions to follow are allocated to predefined areas and the usage of intermediate waypoints makes it possible to build routes that differ from the shortest trajectory to the destination. Moreover, when a constraint favours routes close to obstacles, a potential energy is also assigned to each cell in order to guide the pipe. A cell near an obstacle has a low value while a cell that contains an obstacle has a high value. Then, the optimisation process generates new routes with crossover operations that mix the candidate routes of a generation. The satisfaction of constraints is quantified using a fitness function. At the end of the algorithm, several routes are proposed to the designers who select the best one based on their own experience. In a more recent work, KIMURA [55] proposed an encoding of individuals that describes pipes using the position of their bends and considers also the position of other components, which makes it possible to simultaneously solve the pipe routing and component layout problems. In [92] and [69], multi-branch pipe routing where pipes have more than two terminals is addressed by representing multi-branch pipes as combinations of several two-point pipes. The route between two connections is generated using a MRA and

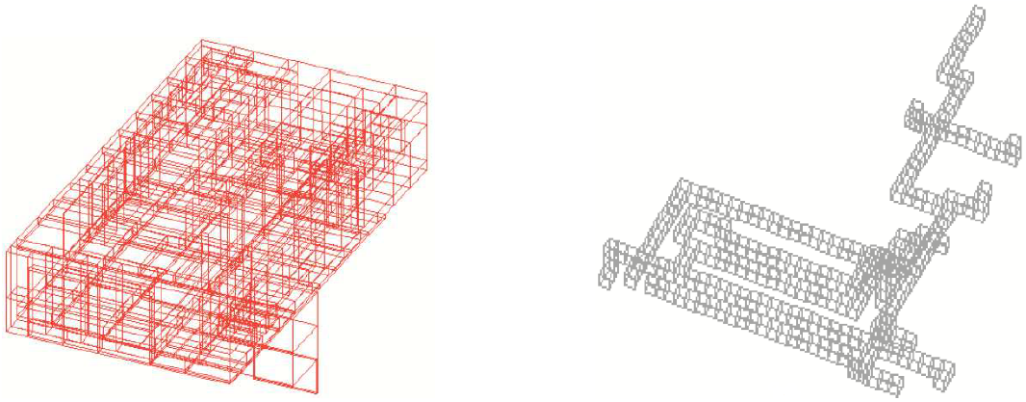
a GA, like Nondominated Sorting Genetic Algorithm II, is used to diversify the multi-branch pipes based on various genetic operators.

Ant Colony Optimisation

Ant Colony Optimisation (ACO) introduced by DORIGO in 1991 [26] has also been experimented to solve the Pipe Routing Problem. In [28] which deals with ship pipe routing, the routing space is decomposed into regular cubic cells and a potential energy is allocated to each cell according to the presence of obstacles or to the distance from preferred routing areas. Then, ACO is performed using individual ants that randomly route the pipe in the adjacency graph of cells. When a solution path is found, a pheromone trail is dropped on its edges depending on the pipe quality considering its total length, its number of bends and its total energy. The quantity of pheromones guides the other ants towards the preferred directions. Pheromones progressively vanish over time and, after a sufficient number of iterations, only the best paths are described by the remaining pheromone trails. More recently, JIANG [47] used Multi-ACO to solve the multi-branch ship pipe routing problem. The convergence of this approach has been improved by launching individual ants from the source and destination cells and by extending the influence area of pheromone trails to neighbouring cells.

Particle Swarm Optimisation

In [5, 4], authors also solve multi-branch pipe routing design in ships by combining the deterministic DIKJSTRA's algorithm with Particle Swarm Optimisation (PSO) introduced by KENNEDY and EBERHART [27, 53]. First, the routing space is split into large cells and pipes are routed using DIKJSTRA's algorithm without taking other pipes into account (see Figure 2.5a). In a second step, processing one pipe at a time, the cells crossed by a pipe are subdivided into new cells with a size corresponding to the diameter of the current pipe and the next pipes are rerouted after removing the cells occupied by previous pipes (see Figure 2.5b). Last, multi-branch pipes for each terminal are connected to routed pipes using the shortest paths given by DIKJSTRA's algorithm between terminals and cells used by existing pipes. The routing order of pipes, the connection order of branch pipes as well as the pipes to preserve or to reroute are defined using PSO.



(a) Decomposition with large cells. (b) Decomposition adapted to pipe diameters.

Figure 2.5 – A two-step cell decomposition.

Mathematical Programming

Recently, BELOV [7] proposed a Constraint Programming (CP) formulation of the multiple pipe routing problem. Each pipe is described using variables which represent the position of its bends in a space discretised with a regular grid. Several complex constraints like safety distances between pipes are taken into account. Commercial solvers like CPLEX allow the resolution of instances with more than 20 pipes within dozens of seconds. To deal with hundreds of pipes, the priority-based search, a multi-agent pathfinding method, has been experimented using BELOV's model as single pipe routing algorithm [8].

2.1.4 Limitations

Most CDAs are based on a grid decomposition of the routing space with a mesh size restricted to be bigger than the pipe diameter. Routing pipes as paths in the adjacency graph of cells assumes that pipes must have axis-parallel segments and use only orthogonal bends. However, in some applications, these assumptions are not acceptable. In particular, when routing waveguides in a telecommunication satellite, space and weight are precious and can be saved by using non orthogonal bends.

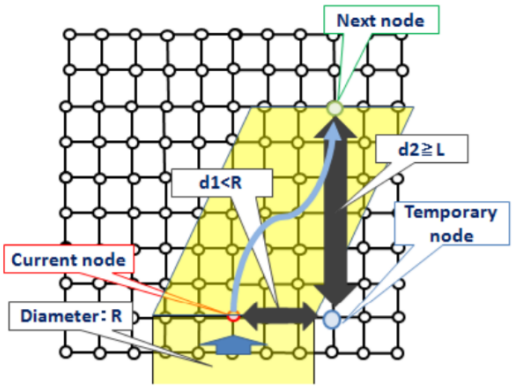


Figure 2.6 – Large neighbourhood.

To address this issue, BESSAIH [9] and ANDO [2] proposed a similar idea. It consists in decomposing the routing space with a mesh size smaller than the waveguide gauge or pipe diameter. A large neighbourhood is then used to define the possible successors of a cell, allowing to reach distant cells that do not require a common boundary with the current cell, as illustrated on Figure 2.6. It results that it is possible to produce non axis-parallel segments and to model non-orthogonal bends depending on the direction selected from a given cell.

However, using a mesh size smaller than the pipe dimension leads to a more complex definition of occupied cells. Indeed, a routed pipe not only crosses the cells that are located on its path in the graph, but it also occupies adjacent cells according to its dimension. In this case, masks can be used to label the occupied cells before routing other pipes, as proposed in [9] (see Figure 2.7). Unfortunately, this approach does not allow to use any non-orthogonal bend angles since each node is still placed on a regular grid. Moreover, some orientation changes generated this way can be non-standard if bends are restricted to a catalogue.

(a) Paths without using masks.

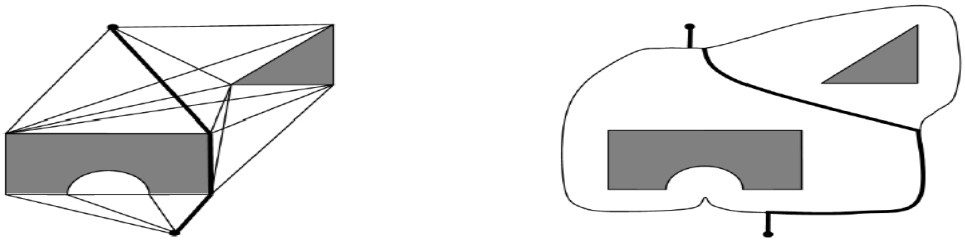
(b) Paths using masks.

Figure 2.7 – Masks for large neighbourhoods.

2.2 Skeleton Approaches

2.2.1 Principle

Another graph-based pipe routing method called Skeleton Approach (SkA) builds a graph of possible route candidates. Then, like in CDA, a pipe solution is a path in the discrete graph, but this time the nodes of the graph are associated with continuous positions that are not forced to be located on a regular grid. There are several ways to create the graph of candidate routes. Visibility graph [42] uses the vertices of obstacles as nodes and contains an edge for each pair of nodes such that the segment connecting them does not intersect any obstacle, as illustrated on Figure 2.8a. Another possibility is VORONOI's diagram that defines a set of points which are equidistant from obstacles [42, 6], as illustrated on Figure 2.8b. Nevertheless, these kinds of skeleton graphs are not suitable to satisfy the constraints on the orientation changes that are involved in Pipe Routing. Indeed, bends along a pipe are often restricted to use specific angles.

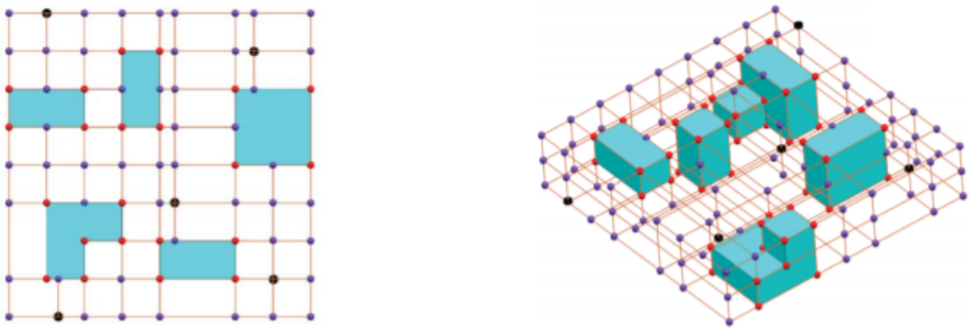


(a) Visibility graph. (b) VORONOI's diagram.

Figure 2.8 – Classical skeleton graphs.

2.2.2 Escape graphs

Another approach to create a skeleton graph consists in using the principle of HIGHTOWER's algorithm [40] (detailed in Section 2.3) in order to build an *escape graph*. To do so, the graph is constructed by extending lines along the horizontal and vertical directions from terminals and obstacle vertices. When the lines encounter an obstacle, a boundary or another line, resulting intersections are added as graph vertices, as shown on Figure 2.9. Escape graphs are widely used in the field of rectilinear path planning [34]. From these graphs, KIM [54] simply proposed to directly apply DIKJSTRA's algorithm to solve the single pipe routing problem.

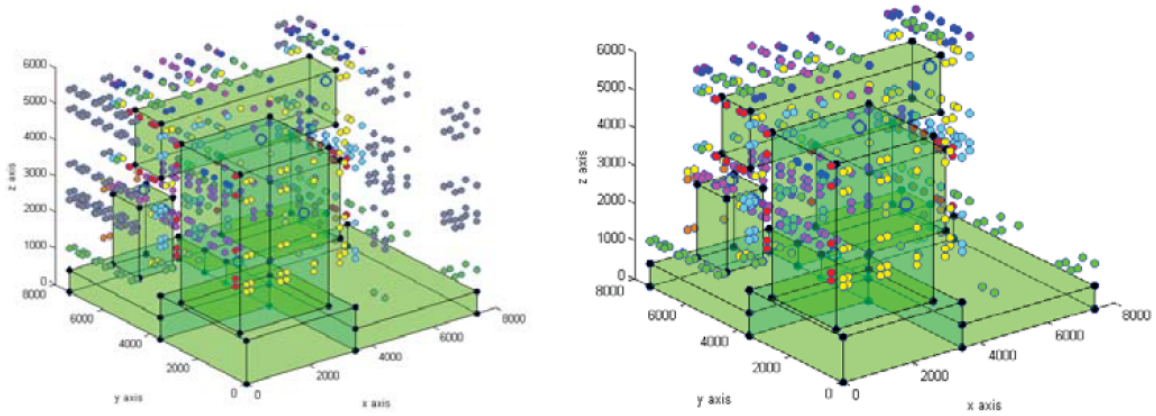


(a) 2D. (b) 3D.

Figure 2.9 – Escape graphs.

In some applications, engineering or safety considerations may require to route pipes close

to the walls or to keep some distance to some obstacles. In those cases, the notion of potential energy related to the distance to obstacles introduced by ITO [44, 45, 46] can be used to reduce the escape graph by deleting nodes with high energy values (see Figure 2.10), that means nodes far from obstacles or in areas to avoid, like in [63]. In this paper, LIU addressed a multi-terminal rectilinear pipe routing problem using a multi-objective evolutionary algorithm based on decomposition and on a representation of individuals with rectilinear STEINER trees. Similar mechanisms of potential energy can be found in many approaches based on skeleton graphs and using various resolution algorithms like ACO [76].



(a) Before deletion. (b) After deletion.

Figure 2.10 – Reducing escape graphs using potential energy.

2.2.3 Graphs inspired by design rules

In [35], GUIRARDELLO builds a graph of candidate routes using a knowledge database inspired by plant layout designers. A node in the graph corresponds to a junction between two portions of route or between two brackets. Edges represent portions of routes or brackets. Nodes said to be primal are placed at the front of each terminal and secondary nodes are created between each pair of primal nodes with perpendicular directions. Then, an edge is added in the skeleton graph between a node and its closest neighbour. Finally, the shortest pipe routes are computed using a MILP formulation or DIKJSTRA's algorithm.

2.2.4 Limitations

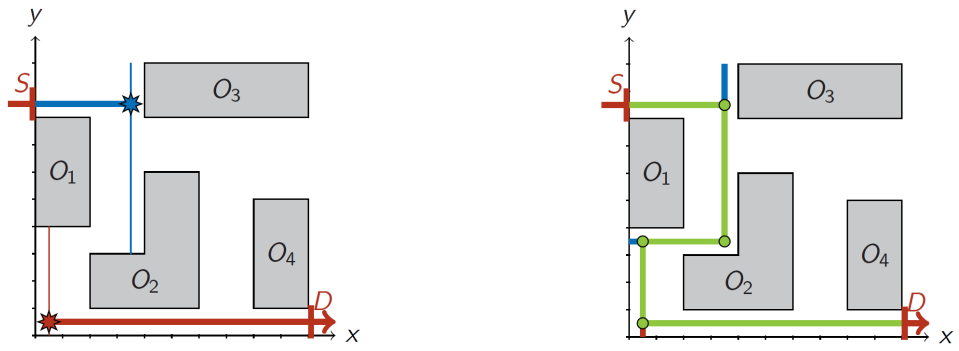
Graph-based approaches are widely used to solve the Pipe Routing Problem (PRP) because they simplify many constraints like conflict avoidance using discretisation. While the Cell Decomposition Approach discretise the environment and the pipes, algorithms based on the Skeleton Approach reduce the possible solutions to a finite set. Nevertheless, in both cases, these methods lead to sub-optimal solutions in the continuous space, which can be restrictive in some industrial situations.

Furthermore, existing graph-based methods assume that pipes have circular or square sections. As a consequence, a pipe is viewed as a succession of regular cells in CDA or as a succession of segments in SkA without considering the pipe cross-section orientation along the route, even if this is necessary with rectangular cross-sections like in WR.

2.3 Line Search Algorithms

2.3.1 Principle

Another way to route pipes in a continuous environment is to use Line Search Algorithms (LSAs), also called Escape Algorithms. This technique, introduced by MIKAMI [65] and HIGHTOWER [40], alternatively extends lines from the origin and destination points using a set of directions and repeats this extension each time a line intersects an obstacle or a boundary, as illustrated on Figure 2.11a. The algorithm terminates as soon as a line from the origin tree meets one coming from the destination (see Figure 2.11b).



(a) Extension. (b) Termination.

Figure 2.11 – Line Search Algorithm.

This method is particularly adapted to VLSI since it is able to provide paths in a continuous plane. It can be run in linear time and consumes less memory-space in comparison with MRAs because it requires to store lines defined by three coordinates rather than working on a grid. Moreover, origin and destination points do not need to be placed on discretised positions, which avoids precision issues. However, the LSA does not guarantee to provide a solution, even if one exists.

2.3.2 Combination with Maze Routing Algorithms

To remedy this drawback, several works proposed to combine the speed of LSAs with the completeness of MRAs. For instance, in [87], SOUKUP performs a LSA towards the destination and, as soon as an obstacle is encountered, LEE's algorithm is applied to circumvent it. Then, the LSA is resumed and explores the space like depth-first search does. This approach has been extended to deal with a 3D-space composed of several plane layers [48] or to consider preferential routing areas like in plant layout [84, 85].

ZHU [104] introduced a similar idea to improve the quality of solutions provided by CDAs. It consists in applying a CDA to define a routing channel and then computing the detailed route inside the channel using a projection procedure close to LSA. Before routing a new pipe, the cells are subdivided around already routed pipes. In case of infeasibility, some routes are backtracked and a mechanism of route protection allows to keep the best routes by preventing them from backtracking.

2.3.3 Limitations

The LSA allows dealing with non orthogonal bends but it does not take into account the orientation of pipe sections, which is required in the case of WR. Indeed, the waveguide must reach

the destination with the right orientation, that is to say with the right angular position of the section around the neutral fibre followed by the barycentre of the section along the pipe. This is not possible with LSAs which only consider the centerline.

2.4 Parametric Models

2.4.1 Principle

The last family of routing methods gathers algorithms based on a parametric description of pipes. Such representations make it possible to provide solutions in a continuous space taking complex design constraints into account, like minimal segment length, bend radius or angles restricted to a catalogue.

2.4.2 Representation as a polyline

A pipe can be defined by its segment lengths and the sequence of angles between the successive segments, as illustrated on Figure 2.12. From this model, routing can be solved with Mathematical Programming techniques. In [64], a route is computed using a Constraint Programming formulation with bend positions in a continuous space as variables. Conflicts with obstacles are avoided through non-overlapping constraints between each pipe segment and obstacles. To reduce the high number of these constraints, obstacles are gathered into clusters.

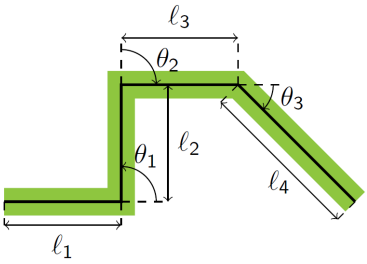


Figure 2.12 – A parametric model of a pipe.

A polyline representation of pipes may also be used in a second routing step in order to improve the solution quality. For instance, VOGEL [98] starts from a reference route provided by DIKJSTRA's algorithm in a grid decomposition of the routing space, and manipulates it adding, removing and/or changing bends using a simulated annealing optimisation scheme.

2.4.3 Representation as blocks

Another possibility consists in modelling a pipe as a set of parallelepipeds that correspond to the volume occupied by the pipe in the routing space, as shown on Figure 2.13. In this case, pipe routing can be addressed simultaneously with component positioning in the spatial packaging problem. SAKTI [83] solves it with a CP formulation that ensures the succession of connector boxes through minimal shared surfaces.

(a) Euclidean geometry. (b) Taxicab geometry.

Figure 2.13 – A solution to the spatial packaging problem.

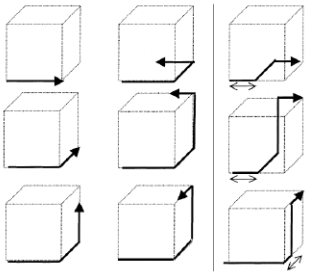
Other methods discretise the set of possible physical pipe components and build pipes iteratively by adding components one by one from a catalogue containing bends and straight

sections of fixed length [31]. However, this method can fail to find feasible solutions since it may be impossible to reach the target with the parts of the catalogue.

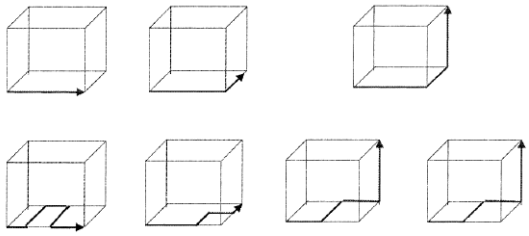
2.4.4 Pattern routing

Last, several approaches construct pipes using predefined patterns with numerical parameters. In [43], authors address the multiple pipe routing problem with a Genetic Algorithm where operators modify a pipe by changing its pattern of generation, that is to say its bend combination, and by increasing or decreasing the lengths of its segments. In this paper, the generated patterns contain up to 5 bends.

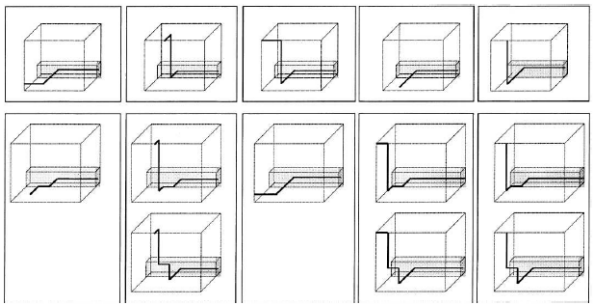
Park [71] proposed a two-step approach to automate routing in a ship engine room. First, candidate routes are provided by a cell generation method based on routing patterns. The routing cells are adapted to the environment and routes are determined during the generation using basic patterns (Figure 2.14a), obstacle avoidance patterns (Figure 2.14b) and connection patterns (Figure 2.14c) inspired from designer experience. Then, a tree search allows selecting the best route for a pipe among several candidates considering additional constraints evaluated using a fitness function.



(a) Basic patterns.



(b) Obstacle avoidance patterns.



(c) Connection patterns.

Figure 2.14 – Routing patterns.

2.5 Motivations of contributions

Most of the previous methods rely on the classical assumptions that the pipe has a circular section, has axis-parallel segments and/or uses orthogonal bends only. These hypotheses are not acceptable for the design of waveguides in a telecommunication satellite for several reasons. Indeed, a satellite payload generally contains many components which make the routing space congested and non orthogonal bends are precious to save space. Moreover, a waveguide is a pipe with a rectangular section so the orientation of its cross-section must be taken into account along the route in order to reach the destination with the right configuration. Furthermore, waveguide

routing approaches proposed by Bessaih [9] ignore several aspects of waveguide design like the usage of standard pieces from a predefined catalogue or the bend radius that is necessary to change the waveguide orientation. For these reasons, no method from the literature can be reused to solve the Waveguide Routing Problem in a sufficiently detailed manner to avoid solution modifications by designers. In this purpose, this thesis introduces new pipe routing techniques that address the three following challenges:

- optimize pipe cost in a 3D continuous routing space;
- use orthogonal and/or non orthogonal bends from a catalogue;
- take into account unsymmetrical pipe sections.

We consider the routing of a single pipe only. In an industrial context, such a routing problem must be solved in a few seconds in order to ensure quick iterations during the design phase. The extended problem which deals with several pipes is left for future work.

Chapter 3

State of the art about Search Algorithms

This thesis proposes formulations of the Waveguide Routing Problem as search problems that are a specific type of computational problems (meaning that a computer might be able to solve them). A search problem is defined by a *search space* $\mathcal{S}$, an *initial state* $s_{ori.}$, a boolean *goal test* referred to as $isGoal\,(s)$ which checks whether a state $s \in \mathcal{S}$ is a target state, and a *successor function* referred to as $successors\,(s)$, also called *neighbourhood* or *transition model*, which maps each state to a set of possible action-successor pairs. The objective of a search problem is to find a sequence of transitions, that is to say a path in search space $\mathcal{S}$, between the initial state $s_{ori.}$ and a goal state respecting the transition model.

Algorithms that solve this kind of problems are called Search Algorithms (SAs) and are one of the primary methods in Artificial Intelligence (AI). They explore *search space* $\mathcal{S}$ in order to find optimal solutions, or good solutions at least, according to a *criterion*. This criterion is classically defined by additive action costs $\gamma\,(s, a, s')$ for applying action $a \in \mathcal{A}$ from state $s \in \mathcal{S}$ leading to state $s' \in \mathcal{S}$, so each path in $\mathcal{S}$ can be mapped to a numeric cost in $\mathbb{R}^+$ to minimise. These algorithms have many application fields. For instance, they are widely used in robotics [62] or in video games for dynamic pathfinding [57].

The efficiency of SAs can be compared through four essential properties: completeness, optimality, time complexity, and space complexity. An algorithm is said to be complete if it guarantees to return a solution if any exists. If the solution returned is guaranteed to be the best among all other solutions according to the criterion, then the algorithm is optimal. Last, the worst-case time complexity describes the maximum computing time needed for an algorithm to complete its task, while the worst-case space complexity is a measure of the maximum memory space required at any point during the search.

Generally, SAs are divided into two families detailed below: uninformed and informed search algorithms. Since a state cannot be generated twice in our application (see Chapter 6 and Chapter 10), the reader should note that the versions of the algorithms presented in what follows do not consider the common issue of avoiding duplicate states (typically, no data structure of already visited states is maintained). Moreover, the complexities exposed here consider searching in an infinite graph.

3.1 Uninformed Search Algorithms

Uninformed Search Algorithms (USAs) do not use any additional knowledge about search space $\mathcal{S}$, such as the closeness to the goal. They operate a brute-force search considering only the initial state $s_{ori.}$, the transition model $successors\,(s)$, and the goal test $isGoal\,(s)$. For these

reasons, they are also called Blind Search Algorithms. The main types of USAs are presented below.

3.1.1 Breadth-First Search

Breadth-First Search (BrFS) is a simple algorithm for searching. It systematically explores the states of search space $\mathcal{S}$ that are reachable from initial state $s_{ori.}$. To do so, the algorithm expands a boundary between discovered and undiscovered states uniformly across the breadth of the boundary. In other words, it discovers all states at level k, which are states that can be reached by applying k actions from $s_{ori.}$, before discovering any state at level $k+1$ [19]. BrFS is detailed in Algorithm 1 and can be efficiently implemented using a queue (First-In First-Out data structure). The algorithm is complete, even if the search space is not finite, but it is optimal only if all action costs are equal. Its time and space complexities are $O\left(b^d\right)$ where d is the depth of the shallowest goal state and b is the *branching factor* or, in other words, the mean number of successors for the states in $\mathcal{S}$. The space requirement of BrFS search is its most critical drawback. BrFS was first used in 1959 by Moore [66] to find paths in mazes. Lee also independently applied the same algorithm in 1961 for wire routing on circuit boards [59] as cited in Chapter 2.

Algorithm 1: Breadth-First Search

Input:

- Initial state: $s_{ori.}$
- Neighbourhood: *successors*
- Goal predicate: *isGoal*

```
1 Boundary ← [s_ori.]
2 while Boundary ≠ ∅ do
3 |   Dequeue state s from Boundary
4 |   if isGoal(s) then
5 |   |   return s
6 |   for (a, s') ∈ successors(s) do
7 |   |   Enqueue state s' into Boundary
8 return NOT_FOUND
```

3.1.2 Depth-First Search

Another approach, called Depth-First Search (DFS), consists in always exploring a descendant of the most recently discovered state that still has unexplored actions until a goal state is reached or until all states have been discovered. Each time all actions of a state have been explored, the algorithm backtracks to the exploration of its parent's actions [20]. A recursive version of DFS is detailed in Algorithm 2.

The time complexity of the algorithm is $O\left(b^d\right)$ where d is the length of the longest path in the search space. However, its space complexity is only $O(bd)$ because only the states on the current path (from the initial state $s_{ori.}$ to the current state s) and their successors must be stored. So, it clearly uses less memory than BrFS. Nevertheless, DFS is not complete if the search space $\mathcal{S}$ is infinite because it can get lost in parts of $\mathcal{S}$ where there is no goal state and never terminate. Morever, it is non-optimal since it can generate solution paths with a high cost to reach a goal state. It has been widely used since the late 1950's, especially in AI programs.

Algorithm 2: Depth-First Search

Input:

- Initial state: $s_{ori.}$
- Neighbourhood: $successors$
- Goal predicate: $isGoal$

```
return depthFirstSearch(s_ori.)
```

Function: $depthFirstSearch\,(s)$
Arguments:

- Current state: s

```
if isGoal(s) then
    return []
for (a, s') ∈ successors(s) do
    Path ← depthFirstSearch(s')
    if Path ≠ NOT_FOUND then
        Push action a in Path
        return Path
return NOT_FOUND
```

3.1.3 Depth-First Iterative Deepening Search

DFS's space-efficiency and BrFS's completeness have been combined by KORF in 1985 [58] with the Depth-First Iterative Deepening Search (DFID). The principle simply consists in performing successive DFS with a limited depth. First, a DFS is performed up to a maximal depth of 1, then the expanded states are discarded and a DFS is performed again with an incremented maximal depth. This process continues until a goal state is reached. So, DFID expands all level k states before expanding states at level $k+1$, like BrFS. This way, it ensures to find the shortest-length solution, if it exists, and this solution will be optimal if all action costs are equal. Even if DFID seems to waste time visiting states multiple times because of restarts, its running time has been shown to be asymptotically in $O\left(b^d\right)$, where d is the depth of the shallowest goal state [58]. Indeed, most states are in the bottom level. Moreover, like DFS, its space complexity is only $O\left(bd\right)$. DFID can be implemented the same way as Iterative Deepening A* Search (IDA*), which is introduced later in Section 3.2.2, by using a zero heuristic $h\left(s\right) = 0$, for any state $s \in \mathcal{S}$, and a unit action cost function $\gamma\left(s, a, s'\right) = 1$, for any action $a \in \mathcal{A}$ leading from $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ (see Algorithm 4).

3.1.4 Uniform Cost Search

It also possible to explore search space $\mathcal{S}$ by always expanding the state with the lowest cost among the ones discovered. This complete approach, called Uniform Cost Search (UCS), is a generalisation of the well known DIJKSTRA's algorithm [24] to infinite search spaces, where all states cannot be enumerated and stored in memory. By definition, the first solution found using UCS is optimal. Its time and space complexities have been proved to be $O\left(b^{1+\left\lfloor \frac{C^*}{\epsilon} \right\rfloor}\right)$, where C^* is the cost of an optimal solution and ϵ is a lower bound on the action costs [81]. It appears that UCS is a particular case of Best-First Search (BFS), described in Section 3.2.1, and its implementation also follows Algorithm 3 using a zero heuristic $h\left(s\right) = 0$ for all $s \in \mathcal{S}$.

3.1.5 Bi-Directional Search

When there is a single explicit goal state and when actions have inverses, it is possible to apply Bi-Directional Search (BDS) [75]. However, in many cases, there are several possible goal states and, even if paths built by backward chaining can be easily defined, the matching between forward and backward states is not always obvious. For these reasons, BDS is not considered here.

3.2 Informed Search Algorithms

Unlike USAs, Informed Search Algorithms (ISAs) use available information from the specified problem and domain knowledge to guide the exploration of search space $\mathcal{S}$. Generally, this information is translated as a heuristic function $h(s)$ that evaluates the closeness of a state $s \in \mathcal{S}$ to the goal. So these algorithms are also called Heuristic Search. They are clearly more efficient than USAs in terms of resolution speed and, for this reason, they are widely used in AI. The main families of ISAs are presented in this section.

3.2.1 Best-First Search

The first major family of ISAs, called Best-First Search (BFS), consists in expanding at each iteration the most promising state according to an evaluation function f. Value $f(s)$ is an estimation of the cost to reach a goal state from the initial state $s_{ori.}$ through state $s \in \mathcal{S}$. BFS is detailed in Algorithm 3 and can be efficiently implemented using a priority queue called *open list* that sorts states according to their evaluation. It is a complete algorithm. Its time complexity depends on the evaluation and/or heuristic functions but the worst-case time complexity is $O\left(b^d\right)$ (if a solution exists). Since it stores all generated states in memory, its space complexity is $O\left(b^d\right)$ which is one of the major drawbacks of BFS.

Algorithm 3: Best-First Search

Input:

- Initial state: $s_{ori.}$
- Neighbourhood: $successors$
- Goal predicate: $isGoal$
- Action cost: γ
- Evaluation: f (with optional heuristic h)

```
1 OpenList ← [s_ori.]
2 while OpenList ≠ ∅ do
3 |   Remove state s with the smallest value f(s) from OpenList
4 |   if isGoal(s) then
5 |   |   return s
6 |   for (a, s') ∈ successors(s) do
7 |   |   g(s') ← g(s) + γ(s, a, s')
8 |   |   Add s' in OpenList with value f(s)
9 return NOT_FOUND
```

In most cases, the evaluation $f(s)$ depends on the best known cost to reach state s from the initial state, referred to as $g(s)$, and/or on a heuristic value $h(s)$ that estimates the remaining

cost to reach a goal state from s. BFS is optimal if f is non-decreasing with depth. The traditional state evaluation function is $f(s) = g(s) + h(s)$ and gives the well known A* Search (A*) introduced by HART in [37] in cases where a state cannot be visited twice. If heuristic $h(s)$ is *admissible*, that is to say if it never overestimates the actual cost to reach a goal state from state s, then A* is guaranteed to return an optimal solution. When heuristic h satisfies the triangular inequality $h(s) \leq \gamma(s, a, s') + h(s')$ for any action $a \in \mathcal{A}$ leading from $s \in \mathcal{S}$ to $s' \in \mathcal{S}$, the heuristic is said to be *consistent*. With a consistent heuristic, A* is optimal and never expands states more than once in cases where states can be revisited [23]. In Weighted A* Search (WA*), the heuristic contribution is accentuated by a weight $\epsilon \geq 1$ which tends to reduce the number of expanded states to find a solution. The solution returned by WA* is guaranteed to cost no more than ϵ times the optimal cost [74]. In the extreme case where $f(s) = h(s)$ and g is completely ignored, called Greedy Best-First Search (Greedy BFS), there is no upper bound on the quality of the solution found [25].

Many sophisticated variants of BFS have been proposed in the literature. A widely used idea is to reduce the size of the relevant search space by defining a subset of promising states and expanding those states only. To do so, the A^*_ϵ algorithm proposed by PEARL [72] uses the open list sorted by evaluation value to select a subset of states to consider for expansion. This subset, called *focal list*, keeps states such that $f(s) < \epsilon f(s_{min})$ where s_{min} is the state having the minimal evaluation. Then, the focal list is sorted by depth and the state with a minimal depth is expanded. Window A* Search also uses state depth to reduce the size of the open list by pruning the current state if the deepest generated state is more than β times deeper [1]. In [30], FURCY proposed the Multi-State Commitment k-Weighted A* Search (MSC-kwA*) algorithm which also maintains a set of privileged states with a given size called *commit list* [30]. At each iteration, k states from this list are expanded and their children are put back into the commit list. As soon as the maximum size of the commit list is reached the worst states are placed in a *reserve list*. These ones are used to refill the commit list when it is not full. Other BFS approaches try to use WA* with weights larger than the user's weight ϵ in order to reduce the number of expanded states while guaranteeing that solutions satisfy the user's upper bound ϵ [94, 93]. Nevertheless, in the recent WILT's study which compares various ISAs [99], WA* and A^*_ϵ are shown to be the most effective BFS approaches. More generally, BFS approaches are competitive because of their completeness on problems where goal states cannot be reached from all states.

In real world problems, resolution time is often limited. In these conditions, it may be interesting to find a solution quickly and then continually work on improving it until time runs out. A procedure following this idea is called an *anytime algorithm*. In this purpose, anytime versions of some BFS approaches can be implemented. For instance, Anytime A* Search (ATA*) proposed in [102] applies successive WA* with decreasing weights ϵ in order to improve the solution found in the previous executions. LIKHACHEV went further and reused the information of the previous WA* by identifying states having inconsistent evaluations each time weight is decreased [62], resulting in the Anytime Repairing A* Search (ARA*).

As previously said, the main drawback of BFS is memory consumption because it requires an exponential space to store the open list. Several variants have been introduced to tackle this problem. The first kind of approaches limits the size of the open list and prunes the least promising states as soon as there is no more available space, like MA* [15], SMA* [82], SMAG* [49] and its revised version [101]. Another possibility is to not maintain the open list in memory. To do so, the same states must be expanded multiple times, but the required space becomes linear with respect to the solution length. Recursive Best-First Search (RBFS) proposed by KORF applies this strategy through recursive calls using a local cost threshold that enables to explore states in best-first order. IDA* presented in Section 3.2.2 is also an example of algorithm

inspired by BFS that does not maintain an open list.

3.2.2 Iterative Deepening A* Search

Iterative Deepening A* Search (IDA*), introduced by KORF in [58], combines the idea of BFS with the low space requirement of DFS. It does not store the open list and executes successive DFS, like in DFID, but prunes paths whose f-evaluations exceed a threshold. The initial threshold is the f-value of the initial node. When the current DFS fails, it updates the boundary to the smallest value that exceeded the previous threshold. IDA* repeats these steps until it finds the optimal solution. Like RBFS, its main drawback is the expansion of the same states multiple times. The process is detailed in Algorithm 4. It only requires to store the states of the current path and their successors, so its space complexity only depends on the solution length and is $O(bd)$ like DFS. Similarly to BFS, its time complexity depends on the heuristic used but is $O\left(b^d\right)$ in the worst-case. Several improvements of IDA* have been proposed, like the exploitation of previously gained state information [77] or the on-line learning of the heuristic [12].

3.2.3 Hill-Climbing

In Hill-Climbing (HC), actions are committed before the search has completed. Typically, basic HC expands one state at a time starting from the initial state and selects only one child of the current state for the next iteration. In Simple Hill-Climbing, the first state closer to the goal is chosen, whereas in Steeple Ascent Hill-Climbing (SAHC) detailed in Algorithm 5 the closest state among all neighbours is selected. Both do not make any effort to guarantee solution quality or completeness. For this reason, HC seems to be extremely competitive in terms of runtime as well as memory required (no information is stored). In fact, such an algorithm very often fail to find a solution since it never backtracks on its decisions and therefore it can fall into areas with no reachable solution, called *dead-ends*. Therefore, HC algorithms are not complete.

Furthermore, HC can also encounter local extrema where the evaluation of the current state is better than the one of its neighbours. In this case, more sophisticated approaches called Enforced Hill-Climbing (EHC) enforce the exploration of worse states in order to find a state closer to the goal. To do so, Stochastic Hill-Climbing randomly selects a neighbour until a new decreasing value is found, which might take a while. A better strategy, introduced by HOFFMANN and NEBEL [41], consists in systematically exploring outwards using BrFS. As soon as a state better than the current one is found, HC resumes its classic procedure.

Real-Time Search (RTS) can be seen as a kind of Hill-Climbing since actions are committed before the search has completed. Although RTS algorithms are designed to satisfy time constraints to take actions in the real world, they can be used to quickly provide solutions to classic search problems. With Local Search Space Learning Real-Time A* (LSS-LRTA*), KOENIG proposed to search from the current state using A* for a fixed number of iterations called *lookahead*, and to commit the best state of the frontier represented by the open list [57]. Then, DIKJSTRA's algorithm is performed to update the heuristic values (this step is not necessary when states cannot be visited twice) before repeating the whole procedure until a goal state is reached or until there is no more child to expand. More recently, improvements have been proposed on LSS-LRTA*, especially to avoid dead-ends [21]. According to WILT's study [99], LSS-LRTA* outperforms other EHCs.

Algorithm 4: Iterative Deepening A* Search

Input:

- Initial state: $s_{ori.}$
- Neighbourhood: $successors$
- Goal predicate: $isGoal$
- Action cost: γ
- Heuristic: h

$Threshold \leftarrow h(s_{ori.})$
$Path \leftarrow [s_{ori.}]$
Loop
  $t \leftarrow search(Path, 0, Threshold)$
  if $t = FOUND$ **then**
    return $(Path, Threshold)$
  if $t = \infty$ **then**
    return NOT_FOUND
  $Threshold \leftarrow t$

Function: $search(Path, g, Threshold)$
Arguments:

- Current path: $Path$
- Cost to reach the current state: g
- Current threshold: $Threshold$

$s \leftarrow Path.getLast()$
$f(s) \leftarrow g(s) + h(s)$
if $f(s) > Threshold$ **then**
  return $f(s)$
if $isGoal(s)$ **then**
  return $FOUND$
$min \leftarrow \infty$
for $s' \in successors(s)$ **do**
  if $s' \notin Path$ **then**
    Push state s' in $Path$
    $t \leftarrow search(Path, g + \gamma(s, s'), Threshold)$
    if $t = FOUND$ **then**
      return $FOUND$
    if $t < min$ **then**
      $min \leftarrow t$
    Remove last state from $Path$
return min

Algorithm 5: Steeple Ascent Hill-Climbing

Input:

- Initial state: $s_{ori.}$
- Neighbourhood: $successors$
- Goal predicate: $isGoal$
- Action cost: γ
- Evaluation: f (with optional heuristic h)

```
1  s_cur. ← s_ori.
2  f_cur. ← ∞
3  while s_cur. ≠ ∅ do
4      if isGoal(s_cur.) then
5          return s_cur.
6      s_cur. ← ∅
7      s_best ← ∅
8      f_best ← ∞
9      for (a, s') ∈ successors(s) do
10         g(s') ← g(s) + γ(s, a, s')
11         if f(s') < f_best then
12             s_best ← s'
13             f_best ← f(s')
14     if f_best < f_cur. then
15         s_cur. ← s_best
16         f_cur. ← f_best
17 return NOT_FOUND
```

3.2.4 Beam Search

The last ISA family considered here is Beam Search (BS). Two kinds of BSs can be distinguished. First, Best-First Beam Search (BF-BS) performs a classic Best-First Search except that the open list has a maximal size [78]. As soon as the size bound is reached, the extra states of lowest quality are removed from the open list. However, states from different depths may be in competition in BF-BS. Shallow states may be advantaged by an admissible heuristic, which ensures that evaluation values $f(s)$ are non-decreasing with depth, although deeper states closer to the goal should be added in the beam instead. It results that BF-BS performs poorly in comparison with Breadth-First Beam Search (BrF-BS), the other type of BSs. This other approach performs a Breadth-First Search except that at each depth only a fixed number of states W, called *beamwidth*, are expanded while the rest of the states is pruned [10]. A simple implementation of BrF-BS is presented in Algorithm 6. It uses a priority queue *OpenList* to sort the neighbours at a given depth by order of evaluation. Since the size of a layer is limited to W, the maximal number of states stored at each iteration corresponds to the maximal size of the open list (in other words, it corresponds to the maximal number of neighbours for a layer) and is $O(bW)$. So beamwidth W allows tuning the amount of memory required by BrF-BS. Moreover, if there are many paths to reach the goal, BrF-BS only considers the W most promising ones. When beamwidth W increases, less promising paths are also considered, which makes it possible to find shallower but potentially more expensive solutions. Last, a larger beam also increases the algorithm runtime because there are more neighbours to enumerate for each layer.

Because of the arbitrary and inadmissible pruning of states, BSs are not complete. However, several variants have been proposed to adapt BrF-BS into complete and anytime algorithms. Typically, it is possible to restart BrF-BS with a wider beam if the previous execution failed, like Complete Anytime Beam Search introduced in [100]. A better idea is to include backtracking, which has been proposed with Beam-Stack Search (BSS) in [103] and Beam search Using Limited discrepancy Backtracking (BULB) in [29]. In practice, a backtrack is translated by selecting another slice of W states rather the W most promising ones to form a layer. To do so, BULB uses the principle of limited discrepancy introduced by HARVEY [38] rather than chronologically backtracking to the parent on the current layer, like Depth-First Beam Search does. This principle consists in making decisions that do not follow the heuristic, called *discrepancies*, at the top of the search tree and iteratively increasing the number of discrepancies while no solution is found.

In WILT's study [99], BSs are shown to be more competitive than BFS in massive search spaces, even if BFS stays more efficient on problems where the goal is not reachable from any state. Last, if the solution must be found within a time limit, BrF-BS is generally preferred to complete BSs because these ones, like BULB and BSS, also require a wide beam to find solutions.

Algorithm 6: Breadth-First Beam Search

Input:

- Beamwidth: W
- Initial state: $s_{ori.}$
- Neighbourhood: $successors$
- Goal predicate: $isGoal$
- Action cost: γ
- Evaluation: f (with optional heuristic h)

```
1  Layer ← [s_ori.]
2  while Layer ≠ ∅ do
3      OpenList ← ∅
4      for s ∈ Layer do
5          if isGoal(s) then
6              return s
7          for (a, s') ∈ successors(s) do
8              g(s') ← g(s) + γ(s, a, s')
9              Add state s in OpenList with value f(s')
10     Layer ← ∅
11     k ← 1
12     while OpenList ≠ ∅ and k ≤ W do
13         Remove s with the smallest f(s) value from OpenList
14         Add state s in Layer
15         k ← k + 1
16 return NOT_FOUND
```

Part II

Waveguide Routing in Free Space

Chapter 4

Free Waveguide Routing Problem

This chapter presents the Free Waveguide Routing Problem (FWRP) which is an optimisation problem consisting in routing a single waveguide in a three-dimensional space free from any obstacle. First, the modelling of a waveguide for the Waveguide Routing Problem (WRP) is presented in Section 4.1 with a generic model usable for any kind of pipe and a simplified model for the WRP. Then, the formalisation of the FWRP is introduced in Section 4.2 and describes the constraints that the waveguide must satisfy as well as the criterion assessing its quality.

4.1 Modelling of a waveguide

The first step in the formalisation of the WRP consists in modelling a waveguide in the three-dimensional canonical Euclidean space $\mathbb{R}^3$. As a reminder, the n-dimensional canonical Euclidean space is the n-dimensional vector or affine real space, that means $\mathbb{R}^n$, provided with the canonical scalar product. A detailed description of any kind of pipe should allow designers to validate its layout according to several aspects. In the case of a waveguide, they must check that it fulfils its function in the RF-harness and that it satisfies some mechanical stress constraints. Furthermore, designers must reserve the space required by the waveguide inside the satellite payload and provide the manufacturing requirements to place order with the waveguide suppliers. To do so, the description should make the computation of the following elements easy:

- the input and output positions and orientations of the waveguide in order to ensure that it is connecting the origin and destination ports;
- the volume occupied by the waveguide to check that it does not collide with any other component of the satellite payload;
- the straight sections and the bends making up the waveguide.

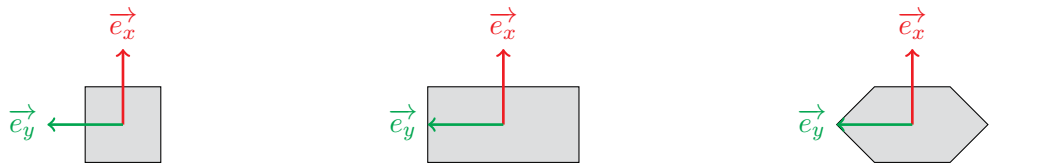
4.1.1 Generic waveguide model

Let Π be the set of the waveguides (or pipes) in $\mathbb{R}^3$. Any waveguide $\pi \in \Pi$ can be described by the following three mathematical objects defined right after:

- a cross-section $\mathcal{S}_\pi$;
- a neutral fibre $\mathcal{F}_\pi$;
- an orientation function o_π.

Definition 1: Cross-section

The *cross-section* of a waveguide $\pi \in \Pi$ is a non-empty bounded polyhedron $\mathcal{S}_\pi$ of the two-dimensional canonical Euclidean space, that means a non-empty convex polygon of $\mathbb{R}^2$ (see Figure 4.1).



(a) Square cross-section (b) Rectangular cross-section (c) Polygonal cross-section

Figure 4.1 – Examples of cross-sections.

As detailed in Chapter 2, in most application fields, the cross-section of the pipes is circular. This is generally the case for pipes which carry a liquid, like water or oil pipelines (see Figure 4.2a). These cross-sections can be approximated by a square, like on Figure 4.1a, or a more complex regular polygon depending on the required accuracy.

(a) Water pipeline parts. (b) Waveguide parts.

Figure 4.2 – Examples of pipe parts.

However, the are many kinds of pipes, like waveguides, with a rectangular cross-section (see Figure 4.2b) or an even more complex one. Such a cross-section has the particularity to be oriented, that means it does not have a centre of symmetry. Therefore, the design of this kind of pipes requires to take the orientation of the cross-section into account.

Definition 2: Neutral fibre

The *neutral fibre* of a waveguide $\pi \in \Pi$ is the curve $\mathcal{F}_\pi$ described by the barycentre of the cross-section $\mathcal{S}_\pi$ along the waveguide as shown on Figure 4.3 on the facing page.

In other words, the neutral fibre $\mathcal{F}_\pi$ is the trajectory of waveguide π.

From a cross-section $\mathcal{S}_\pi$ and a neutral fibre $\mathcal{F}_\pi$, the last element required to completely describe a waveguide $\pi \in \Pi$ is to define the orientation of the cross-section at each point $P \in \mathcal{F}_\pi$ of the neutral fibre.

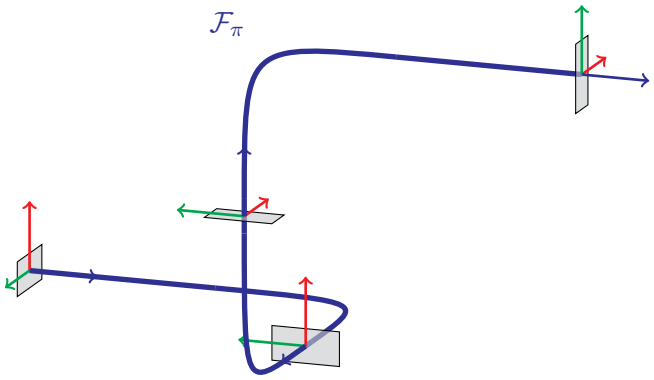


Figure 4.3 – Example of neutral fibre $\mathcal{F}_\pi$ (in blue) for a waveguide $\pi \in \Pi$.

Definition 3: Orientation

An *orientation* is an orthonormal basis $o = (\overrightarrow{e_{o,x}}, \overrightarrow{e_{o,y}}, \overrightarrow{e_{o,z}})$ of the canonical Euclidean space $\mathbb{R}^3$. The set of orientations is referred to as $\mathcal{O}$ and o_{ref} is the reference orientation in $\mathbb{R}^3$. Moreover, in what follows, M_o is the 3D rotation matrix which expresses the orientation o in the reference orientation o_{ref}.

Definition 4: Orientation function

The *orientation function* of a waveguide $\pi \in \Pi$ is the function o_π defined by:

$$\begin{array}{rccc} o_\pi & : & \mathcal{F}_\pi & \to & \mathcal{O} \\ & & P & \mapsto & o_\pi(P) \end{array}$$

such that (see Figure 4.4):

- $\overrightarrow{e_{o_\pi(P),z}}$ is tangential to the neutral fibre $\mathcal{F}_\pi$ of waveguide π at point P;
- the intersection between the plane of normal $\overrightarrow{e_{o_\pi(P),z}}$ passing through point P with the volume of waveguide π corresponds to cross-section $\mathcal{S}_\pi$ expressed in the frame $\left(P, \overrightarrow{e_{o_\pi(P),x}}, \overrightarrow{e_{o_\pi(P),y}}\right)$ of $\mathbb{R}^2$.

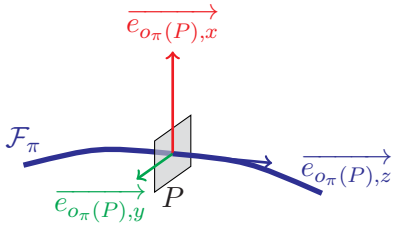


Figure 4.4 – Orientation $o_\pi(P)$ at a point $P \in \mathcal{F}_\pi$ of a waveguide $\pi \in \Pi$.

When there is a discontinuity on the tangent of neutral fibre $\mathcal{F}_\pi$ at a break point $P_{disc.} \in \mathcal{F}_\pi$, only the orientation after the break point is considered:

$$o_\pi(P_{disc.}) = \lim_{P \to P^+_{disc.}} o_\pi(P)$$

Definition 5: Configuration

A *configuration* is an orthonormal frame $\theta = (P_\theta, \overrightarrow{e_{\theta,x}}, \overrightarrow{e_{\theta,y}}, \overrightarrow{e_{\theta,z}})$ of the canonical Euclidean space $\mathbb{R}^3$ (see Figure 4.5). The set of configurations is referred to as Θ and θ_{ref} is the reference configuration in $\mathbb{R}^3$.

Definition 6: Configuration function

By analogy with the orientation function, the *configuration function* of a waveguide $\pi \in \Pi$ is the function θ_π defined by:

$$\begin{array}{rccc} \theta_\pi & : & \mathcal{F}_\pi & \to & \Theta \\ & & P & \mapsto & \theta_\pi(P) = (P, o_\pi(P)) \end{array}$$

4.1.2 Simplified waveguide model

In practice, engineers design waveguides as a succession of straight sections and bends. Moreover, this succession also corresponds to the manufacturing process of waveguides, as explained in Section 1.2.1. The previous modelling of a waveguide can be simplified by defining its straight sections and bends. To do so, the definitions of a translation and a rotation of a configuration are introduced.

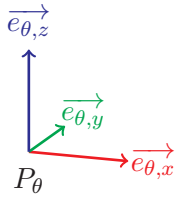


Figure 4.5 – A configuration $\theta \in \Theta$.

Definition 7: Translation

A *translation* of length $L \in \mathbb{R}$ is the function tr_L defined by:

$$\begin{array}{rccc} tr_L & : & \Theta & \to & \Theta \\ & & \theta = (P, o) & \mapsto & (P + L \cdot \overrightarrow{e_{o,z}}, o) \end{array}$$

Definition 8: Rotation

A *rotation* of 3D matrix M is the function rot_M defined by:

$$\begin{array}{rccc} rot_M & : & \Theta & \to & \Theta \\ & & \theta = (P, o) & \mapsto & (P, o') \end{array}$$

where o' is the orientation defined by the rotation matrix $M_{o'} = MM_o$.

Definition 9: Straight section

A *straight section* is a function u such that there is a positive real $L_u \in \mathbb{R}^+$, called *length* of the straight section, which verifies (see Figure 4.6 on the next page):

$$\begin{array}{rccc} u & : & \Theta & \to & \Theta \\ & & \theta & \mapsto & u(\theta) = tr_{L_u}(\theta) \end{array}$$

The set of straight sections is referred to as $\mathcal{U}$ and, for a positive real $L \in \mathbb{R}^+$, u_L is the straight section of length L.

For a straight section $u \in \mathcal{U}$ applied from a configuration $\theta \in \Theta$, such as $\theta = (P, o)$, the following properties are verified:

- its neutral fibre, referred to as $\mathcal{F}_{u,\theta}$, is segment $\left[P, P_{u(\theta)}\right]$;
- its orientation function, referred to as $o_{u,\theta}$, is constant with $o_{u,\theta}(P') = o$ for any point P' on neutral fibre $\mathcal{F}_{u,\theta}$.

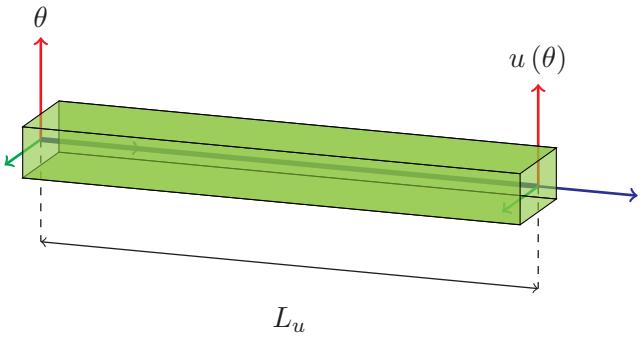


Figure 4.6 – A straight section $u \in \mathcal{U}$ applied from configuration $\theta \in \Theta$.

Definition 10: Bend

A *bend* is a function b such that there is a positive real $L_b \in \mathbb{R}^+$, called *half-length* of the bend, and a rotation matrix M_b which verify (see Figure 4.7):

$$\begin{array}{rccc} b & : & \Theta & \rightarrow & \Theta \\ & & \theta & \mapsto & b(\theta) = tr_{L_b} \circ rot_{M_b} \circ tr_{L_b}(\theta) \end{array}$$

The set of bends is referred to as $\mathcal{B}$.

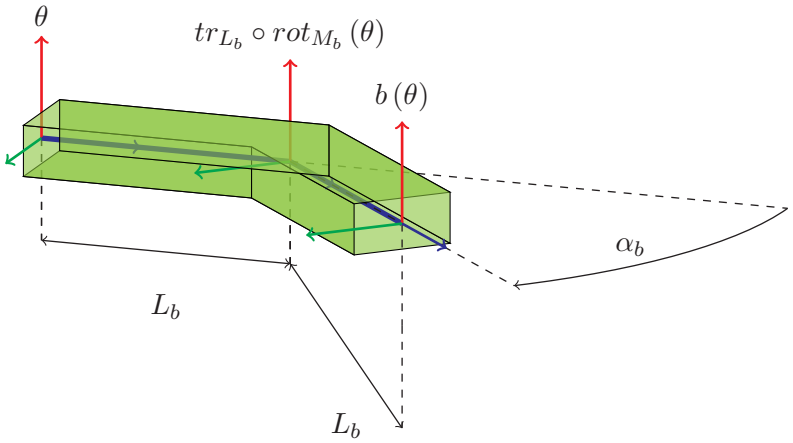


Figure 4.7 – A bend $b \in \mathcal{B}$ applied from configuration $\theta \in \Theta$.

In pratice, for a bend $b \in \mathcal{B}$, rotation matrix M_b corresponds to a rotation around a canonical axis, that means the axis $\overrightarrow{x}$, $\overrightarrow{y}$ or $\overrightarrow{z}$ of the local frame. In the case of waveguides, the *angle* of the rotation, referred to as α_b, is never obtuse, what can be translated as:

$$\alpha_b \in \left[\frac{-\pi}{2}, \frac{\pi}{2}\right]$$

This is mainly due to manufacturing constraints. For instance, it is not possible to bend a straight section with an obtuse angle in order to create a formed bend. Furthermore, it is possible to define a *bend radius* $\rho_b \in \mathbb{R}^+$ such that (see Figure 4.8):

$$L_b = \rho_b \tan\left(\frac{\alpha_b}{2}\right)$$

Then, for a bend $b \in \mathcal{B}$ applied from a configuration $\theta \in \Theta$, such that $\theta = (P, o)$ and $o' \in \mathcal{O}$ is the orientation defined by the rotation matrix $M_{o'} = M_b M_o$, the following properties are verified:

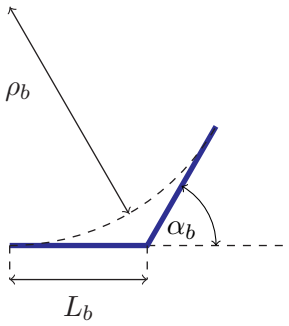


Figure 4.8 – Relation between radius ρ_b, angle α_b and half-length L_b.

- its neutral fibre, referred to as $\mathcal{F}_{b,\theta}$, is the polyline $\left[P, P_{tr_{L_b}(\theta)}, P_{b(\theta)}\right]$;
- its orientation function, referred to as $o_{b,\theta}$, is piecewise constant with $o_{b,\theta}(P') = o$ for any point $P' \in \left[P, P_{tr_{L_b}(\theta)}\right[$ and $o_{b,\theta}(P') = o'$ for any point $P' \in \left[P_{tr_{L_b}(\theta)}, P_{b(\theta)}\right]$.

From the previous definitions of straight sections and bends, the formalisation of a waveguide can be simplified.

Definition 11: Waveguide

Let N_π be a positive integer and $B_{cat.} \subset \mathcal{B}$ a subset of bends. A *waveguide* is a couple $\pi = (\theta_\pi^{ori.}, \sigma_\pi)$ made up with an *origin configuration* $\theta_\pi^{ori.} \in \Theta$ and a composition σ_π alternating straight sections $u_{\pi,k} \in \mathcal{U}$, for $k \in [\![1, N_\pi]\!]$, and bends $b_{\pi,k} \in B_{cat.}$, for $k \in [\![1, N_\pi - 1]\!]$ (see Figure 4.9 on the facing page). In other words, the composition σ_π of waveguide π can be written as follows:

$$\sigma_\pi = u_{\pi,N_\pi} \circ b_{\pi,N_\pi-1} \circ u_{\pi,N_\pi-1} \circ ... \circ u_{\pi,2} \circ b_{\pi,1} \circ u_{\pi,1}$$

The set of waveguides using bend catalogue $B_{cat.}$ (see Section 1.2.1) is referred to as $\Pi(B_{cat.})$.

So, for a waveguide $\pi \in \Pi$, the following properties can be proved by induction:

- its neutral fibre $\mathcal{F}_\pi$ is a polyline $[P_{\pi,1}, ..., P_{\pi,N_\pi+1}]$ composed of N_π segments;
- its orientation function o_π is piecewise constant: for $k \in [\![1, N_\pi]\!]$, there is an orientation $o_{\pi,k} \in \mathcal{O}$ such that $o_\pi(P) = o_{\pi,k}$ for any point $P \in [P_{\pi,k}, P_{\pi,k+1}[$. By convention, $o_\pi(P_{\pi,N_\pi+1}) = o_{\pi,N_\pi}$.

Thus, the usage of this simplified definition of a waveguide consists in approximating the neutral fibre of real waveguides, which can have curved parts, by a polyline such that the cross-section orientation is constant along the segments. Moreover, the transition from an orientation to the one of the next segment depends on the bend applied at the end of the first segment.

Definition 12: Orientation change

An *orientation change* is a tuple $r = (o_r^-, o_r^+, b_r)$ where $o_r^-, o_r^+ \in \mathcal{O}$ are orientations and $b_r \in \mathcal{B}$ is a bend which verifies:

$$o_r^+ = rot_{M_{b_r}}(o_r^-)$$

The set of orientation changes is referred to as $\mathcal{R}$.

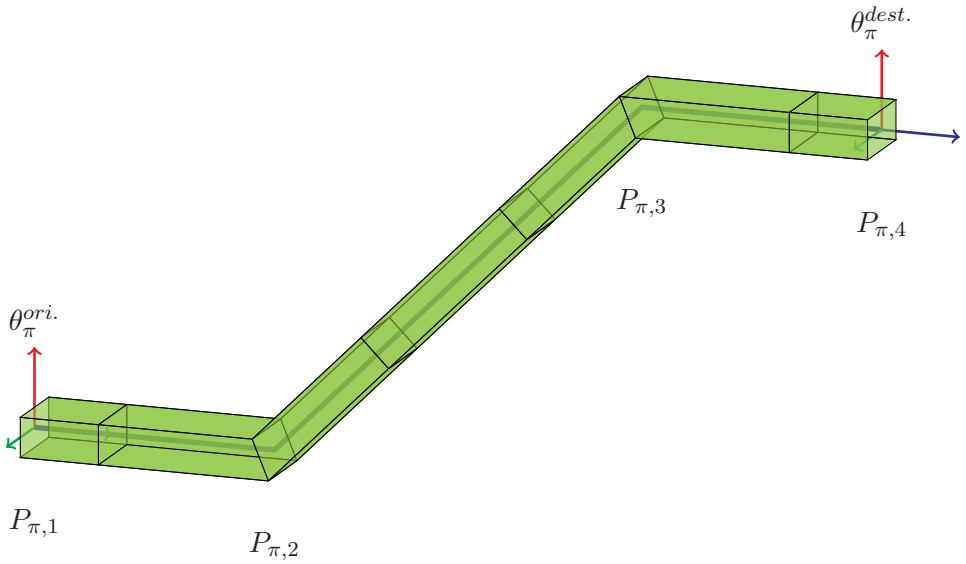


Figure 4.9 – A waveguide $\pi \in \Pi$.

In what follows, for a waveguide $\pi \in \Pi$, the following notations are used:

- by analogy with $\theta_\pi^{ori.}$, the destination configuration of waveguide π is referred to as $\theta_\pi^{dest.}$ and satisfies:
$$\theta_\pi^{dest.} = \sigma_\pi\left(\theta_\pi^{ori.}\right)$$
- $\ell_{\pi,k}$ is the length of the k^{th} segment of the neutral fibre $\mathcal{F}_\pi$, that is $[P_{\pi,k}, P_{\pi,k+1}]$, for $k \in [\![1, N_\pi]\!]$;
- $r_{\pi,k}$ is the orientation change applied at the end of the k^{th} segment of the neutral fibre $\mathcal{F}_\pi$, for $k \in [\![1, N_\pi - 1]\!]$.

4.2 Definition of the FWRP

The Free Waveguide Routing Problem (FWRP) consists in routing in a detailed manner a single waveguide in a free three-dimensional space. In other words, the goal is to find a waveguide $\pi \in \Pi$ which connects an origin polyhedron $\mathcal{P}^{ori.}$ to a destination polyhedron $\mathcal{P}^{dest.}$ without considering any obstacle nor routing space restrictions. However, the solution waveguide π must satisfy the specific constraints presented in Section 4.2.1 coming from the design and manufacturing rules of the RF-harness (see Section 1.2.1). At the same time, the quality of waveguide π should be as good as possible, which can be translated as the minimisation of the cost criterion defined in Section 4.2.2. Finally, Section 4.2.3 sums up the constraints and criterion in order to formulate the full FWRP problem.

4.2.1 Constraints

Connectivity

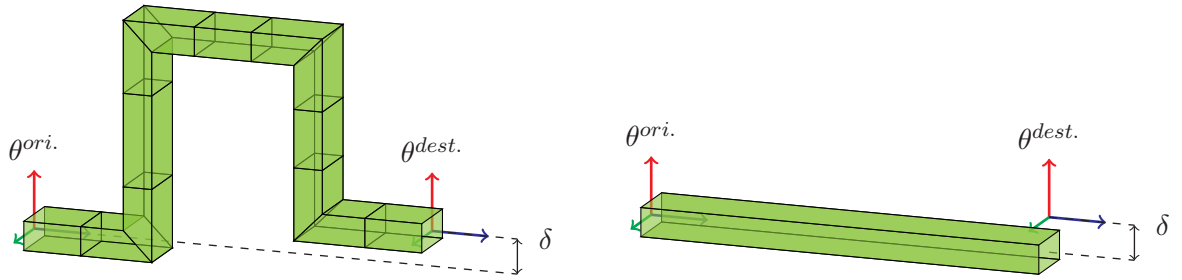
First of all, waveguide $\pi \in \Pi$ must ensure the connection between its origin and destination ports in order to fulfil its mission inside the RF-harness. To do so, the first and last points of

its neutral fibre $\mathcal{F}_\pi$ should be respectively contained in the convex polyhedrons $\mathcal{P}^{ori.}$ and $\mathcal{P}^{dest.}$, which can be formulated as:

$$P_{\pi,1} \in \mathcal{P}^{ori.} \tag{4.1}$$

$$P_{\pi,N_\pi+1} \in \mathcal{P}^{dest.} \tag{4.2}$$

Using polyhedrons $\mathcal{P}^{ori.}$ and $\mathcal{P}^{dest.}$ instead of fixed origin and destination points allows keeping some flexibility in early design phases of the RF-harness. For instance, if a straight section can connect the origin and destination ports with a small position error $\delta > 0$, it can be relevant to slightly move the origin or destination point instead of using a complex waveguide that contains several bends or even a loop to compensate for the position error δ, as shown on Figure 4.10.



(a) Without tolerance. (b) With tolerance.

Figure 4.10 – Tolerance $\delta > 0$ on the destination position.

Furthermore, waveguide π has to respect the orientation of the cross-section at each end to be plugged on the origin and destination ports. So, the orientation of the first segment of waveguide π is restricted to a unique origin orientation $o^{ori.}$, whereas at the destination there exists a set of possible destination orientations $\mathcal{O}^{dest.}$, either because the destination orientation is still flexible or because some orientations are equivalent from a cross-section point of view (see Figure 4.11). Thus, the orientation constraints which should be verified are:

$$o_{\pi,1} = o^{ori.} \tag{4.3}$$

$$o_{\pi,N_\pi} \in \mathcal{O}^{dest.} \tag{4.4}$$

Figure 4.11 – Example of equivalent orientations for a rectangular cross-section.

Bend catalogue

For the manufacturing and economical reasons explained in Section 1.2.1, all bends used in waveguide π should be taken from a catalogue $B_{cat.} \subset \mathcal{B}$. Recall that, in practice, waveguide manufacturers generally provide catalogues which define the possible bends according to the

cross-section $\mathcal{S}_\pi$ (see Figure 4.12). Moreover, using a bend catalogue maximises the reuse of each bend type. The catalogue constraint can be written as follows:

$$\pi \in \Pi\left(B_{cat.}\right) \tag{4.5}$$

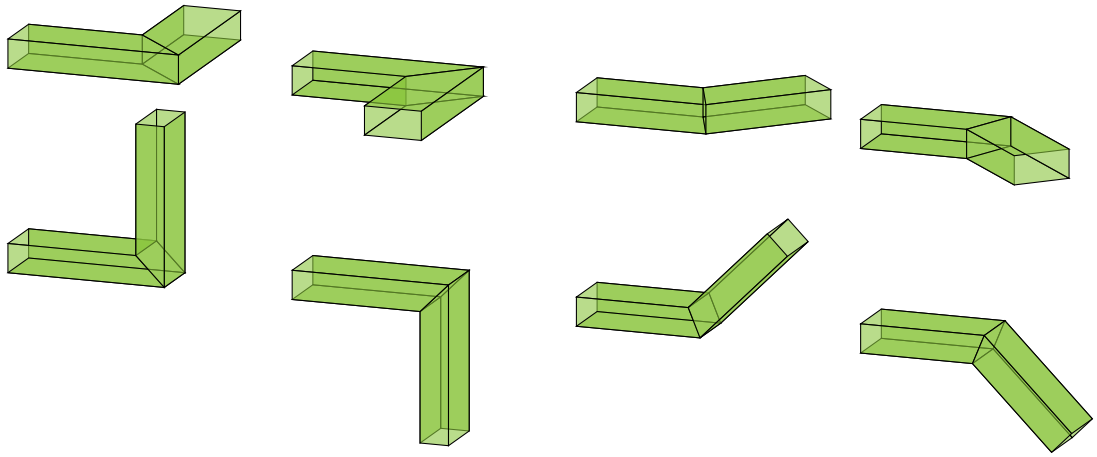


Figure 4.12 – A bend catalogue with 90°-bends and 45°-bends.

Maximum number of bends

Like in most Pipe Routing application fields (see Chapter 2), it is often profitable to minimise the number of bends used in waveguide π. Indeed, bends generate additional radio-frequency losses on the electromagnetic signal carried by the waveguide. Therefore, waveguide π has to use a maximum number of segments $N_S \geq 1$ in order to limit the impact of bends on the quality of the waveguide:

$$N_\pi \leq N_S \tag{4.6}$$

Minimum length of straight sections

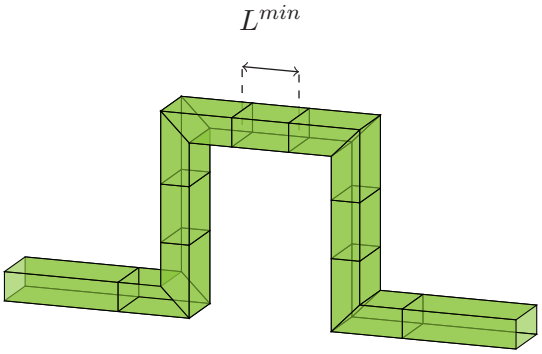


Figure 4.13 – Minimal straight length L^{min} between two bends.

As presented in Section 1.2.1, there are two kinds of waveguide bends: the formed bends and the machine ones. Both require a straight section with minimal length $L^{min} \in \mathbb{R}^+$ between two successive bends, as shown on Figure 4.13. For this reason, any straight section between two bends of waveguide π should respect the minimum length L^{min} given by manufacturers:

$$\forall k \in [\![1, N_\pi]\!] \qquad L^{min} \leq L_{u_{\pi,k}} \tag{4.7}$$

Global attachability

As explained in Section 1.2.1, waveguide π must be fixed on a panel of the satellite. However, when the bend catalogue contains non-orthogonal bends, there are some bend combinations that systematically lead to segments with unattachable orientations regardless of the wall on which they are routed (see Figure 4.14). Such waveguides are said to be *infeasible* and waveguide designers refrain using them in order to satisfy the manufacturing and attachability constraints.

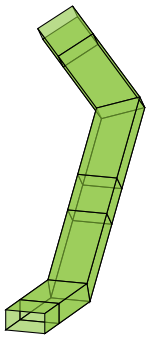


Figure 4.14 – An infeasible waveguide.

To avoid infeasible waveguides, the orientation of any segment of waveguide π must have a globally attachable orientation.

Definition 13: Global attachability

The set of *globally attachable orientations* is referred to as $\mathcal{O}^{int.} \subset \mathcal{O}$ and an orientation $o \in \mathcal{O}$ is said to be *globally attachable* if:

$$o \in \mathcal{O}^{int.}$$

So, for waveguide π, the *global attachability constraints* can be translated as:

$$\forall k \in [\![1, N_\pi]\!] \qquad o_{\pi,k} \in \mathcal{O}^{int.} \tag{4.8}$$

In practice, the set of globally attachable orientations depends only on the origin orientation $o^{ori.}$ of waveguide π. Indeed, one can ensure that waveguide π is feasible if all its segments have an orientation $o_{\pi,k}$ such that either $\overrightarrow{e_{o_{\pi,k},x}}$ or $\overrightarrow{e_{o_{\pi,k},y}}$ is orthogonal to $\overrightarrow{e_{o^{ori.},z}}$, for $k \in [\![1, N_\pi]\!]$. Thus, globally attachable orientations can be defined by:

$$\mathcal{O}^{int.} = \left\{ o \in \mathcal{O} \;\middle|\; \left(\overrightarrow{e_{o,x}} \cdot \overrightarrow{e_{o^{ori.},z}} = 0\right) \vee \left(\overrightarrow{e_{o,y}} \cdot \overrightarrow{e_{o^{ori.},z}} = 0\right) \right\} \tag{4.9}$$

Assumption 1:

Is is assumed that the origin orientation and the destination orientations are globally attachable:

$$o^{ori.} \in \mathcal{O}^{int.} \qquad\qquad \mathcal{O}^{dest.} \subseteq \mathcal{O}^{int.}$$

4.2.2 Criterion

The cost of waveguide π is formulated as a function γ_π to be minimised. This cost can represent several aspects of the RF-harness design, like manufacturing price or radio-frequency losses along the waveguide. Generally, it depends on the length of waveguide π, given a linear cost $\mu \in \mathbb{R}^+$, and on the bends used in the waveguide, given that each bend $b \in \mathcal{B}$ has a unit cost $\gamma_b \in \mathbb{R}^+$. The lower the cost γ_π, the better the quality of waveguide π.

Definition 14: Cost

The *cost* of a waveguide $\pi \in \Pi$ is defined by:

$$\gamma_\pi = \sum_{k=1}^{N_\pi - 1} \gamma_{b_{\pi,k}} + \mu \sum_{k=1}^{N_\pi} \ell_{\pi,k}$$

where $\mu \in \mathbb{R}^+$ is a linear cost and $\gamma_b \in \mathbb{R}^+$ is the unit cost of bend $b \in \mathcal{B}$.

One way to interpret the unit cost γ_b of a bend $b \in \mathcal{B}$ relatively to the linear cost μ is that it is preferable to use bend b if it allows reducing the total waveguide length by at least $\frac{\gamma_b}{\mu}$.

4.2.3 Full FWRP model

To sum up, the FWRP is an optimisation problem that can be written as follows:

$$\begin{aligned}
&\text{minimise } \gamma_\pi = \textstyle\sum_{k=1}^{N_\pi - 1} \gamma_{b_{\pi,k}} + \mu \sum_{k=1}^{N_\pi} \ell_{\pi,k} \\
&\text{subject to:} \\
&P_{\pi,1} \in \mathcal{P}^{ori.} && \text{Connectivity (position at origin)} \\
&P_{\pi,N_\pi+1} \in \mathcal{P}^{dest.} && \text{Connectivity (position at destination)} \\
&o_{\pi,1} = o^{ori.} && \text{Connectivity (orientation at origin)} \\
&o_{\pi,N_\pi} \in \mathcal{O}^{dest.} && \text{Connectivity (orientation at destination)} \\
&\pi \in \Pi\left(B_{cat.}\right) && \text{Bend catalogue} \\
&N_\pi \leq N_S && \text{Maximum number of bends} \\
&L^{min} \leq L_{u_{\pi,k}} \quad \forall k \in [\![1, N_\pi]\!] && \text{Minimal length of straight sections} \\
&o_{\pi,k} \in \mathcal{O}^{int.} \quad \forall k \in [\![1, N_\pi]\!] && \text{Global attachability}
\end{aligned}$$

Chapter 5

Resolution of the FWRP using Mixed Integer Linear Programming

The first approach explored to solve the FWRP is Mixed Integer Linear Programming (MILP). It is an exact or complete method for combinatorial optimisation, that systematically explores the search space and guarantees the optimality of the solution when it terminates. In order to reduce the possible orientations of the waveguide segments to a finite set, the orientation space $\mathcal{O}$ requires to be enumerated during a preprocessing step, detailed in Section 5.1. Then, it is possible to formulate the FWRP as a MILP model presented in Section 5.2. The results of experiments on test instances are shown in Section 5.3.

5.1 Input preprocessing

The formalisation of the FWRP as a MILP model requires variables characterising the vertices of neutral fibre $\mathcal{F}_\pi$ and the length and orientation of each of the N_S possible segments of $\mathcal{F}_\pi$. Intuitively, only a subset of orientations from $\mathcal{O}$ can be reached from the origin orientation $o^{ori.}$ using exactly $k \in [\![1, N_S - 1]\!]$ bends from catalogue $B_{cat.}$ and satisfying the global attachability constraints defined by $\mathcal{O}^{int.}$. So the first step to build a MILP model of the FWRP consists in enumerating all reachable orientations in the form of a graph $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$ called *kernel of reachable orientations* (see Section 5.1.1). Then, using minimal bend combinations introduced in Section 5.1.2 to reach a destination orientation in $\mathcal{O}^{dest.}$, it is possible to reduce the candidate orientations for each segment by building an directed acyclic graph $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$ called *space of candidate orientations* (see Section 5.1.3).

In what follows, for a rotation matrix M, rot_M refers indistinctly to the rotation of matrix M in the configuration space Θ or in the orientation space $\mathcal{O}$.

5.1.1 Kernel of reachable orientations

Definition 15: Reachable orientation

A *reachable orientation* is an orientation $o \in \mathcal{O}^{int.}$ which can be reached from the origin orientation $o^{ori.}$ using bends from catalogue $B_{cat.}$, which can be written as:

$$\exists k \in \mathbb{N} \quad \exists (b_1, ..., b_k) \in B_{cat.}^k \qquad o = rot_{M_{b_k}} \circ ... \circ rot_{M_{b_1}} \left(o^{ori.}\right)$$

The set of reachable orientations is referred to as $\mathcal{O}_\infty$.

Definition 16: Reachable orientation change

A *reachable orientation change* is an orientation change $r \in \mathcal{R}$ which verifies:

$$o_r^- \in \mathcal{O}_\infty \qquad o_r^+ \in \mathcal{O}_\infty \qquad b_r \in B_{cat.}$$

The set of reachable orientation changes is referred to as $\mathcal{R}_\infty$.

By definition, if the instance of the FWRP has a solution, then at least one destination orientation $o \in \mathcal{O}^{dest.}$ is a reachable orientation, which can be formally expressed as:

$$\exists o \in \mathcal{O}^{dest.} \qquad o \in \mathcal{O}_\infty$$

Otherwise, there is no bend combination which allows reaching a destination orientation and the FWRP has no solution.

Global attachability constraints introduced in Section 4.2.1 on page 48 to avoid infeasible waveguides and ensure attachability are restrictive. It can be empirically shown that the set of reachable orientations $\mathcal{O}_\infty$ is finite (see Section 5.3.2 on page 64). In order to enumerate these orientations, the graph $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$, called *kernel of reachable orientations*, is generated using Algorithm 7. It successively applies bends until no more unknown orientation is generated.

Algorithm 7: Generate the kernel of reachable orientations $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$

Input:

- Origin orientation: $o^{ori.}$
- Bend catalogue: $B_{cat.}$
- Global attachable orientations: $\mathcal{O}^{int.}$

```
1  O_∞ ← {o^ori.}
2  OpenList ← {o^ori.}
3  while OpenList ≠ ∅ do
4      Remove o from OpenList
5      for b ∈ B_cat. do
6          o' ← rot_{M_b}(o)
7          if o' ∈ O^int. then
8              if o' ∉ O_∞ then
9                  Add node o' in O_∞
10                 Add node o' in OpenList
11             Add edge (o, o', b) in R_∞
12 return G(O_∞, R_∞)
```

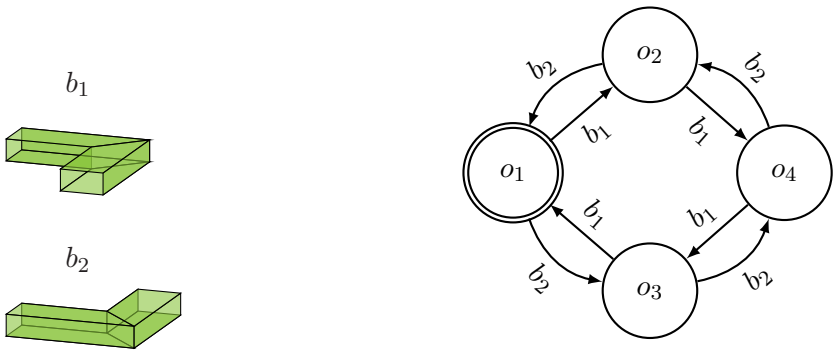
In the sections that follow, $\mathcal{R}_\infty^{inc.}(o)$ and $\mathcal{R}_\infty^{out.}(o)$ refer respectively to the incoming and outgoing reachable orientation changes into/from a reachable orientation $o \in \mathcal{O}_\infty$.

The generation of kernel $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$ is illustrated by a small example that will be reused later. Let the globally attachable orientations be defined by $\mathcal{O}^{int.} = \{o_1, o_2, o_3, o_4\}$ and the bend

catalogue by $B_{cat.} = \{b_1, b_2\}$ (see Figure 5.1a) such that:

$$\begin{aligned} rot_{M_{b_1}}(o_1) &= o_2 & rot_{M_{b_2}}(o_1) &= o_3 \\ rot_{M_{b_1}}(o_2) &= o_4 & rot_{M_{b_2}}(o_2) &= o_1 \\ rot_{M_{b_1}}(o_3) &= o_1 & rot_{M_{b_2}}(o_3) &= o_4 \\ rot_{M_{b_1}}(o_4) &= o_3 & rot_{M_{b_2}}(o_4) &= o_2 \end{aligned}$$

Figure 5.1b shows the kernel of reachable orientations $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$ generated for $o^{ori.} = o_1$.



(a) Bend catalogue $B_{cat.}$. (b) Reachable orientation kernel $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$.

Figure 5.1 – Reachable orientation kernel $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$ generated by Algorithm 7.

The global attachability constraints are relative to the origin orientation $o^{ori.}$ (see Section 4.2.1). For an efficient implementation, the reachable orientation kernel $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$ can be computed only once for a given catalogue $B_{cat.}$ and expressed relatively to the reference orientation o_{ref} given global attachability constraints $\mathcal{O}^{int.}$. Then, the kernel $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$ dedicated to an instance of the FWRP using the same bend catalogue and the same global attachability constraints is deduced by applying a change of basis from $o^{ori.}$ to o_{ref} to all the orientations of the reference reachable orientation kernel. This way, the reuse of the results from previous instances is maximised.

5.1.2 Minimal bend combinations

From the reachable orientation kernel $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$, it is possible to evaluate the minimal bend combination to reach a destination orientation from any reachable orientation $o \in \mathcal{O}_\infty$ given a cost function $\gamma(r)$ defined for any reachable orientation change $r \in \mathcal{R}_\infty$. To do this, Algorithm 8 on the next page applies DIJKSTRA's algorithm [24] from the destination orientations of $\mathcal{O}^{dest.}$ to evaluate the minimal costs $\gamma(o)$ needed to reach them from any orientation $o \in \mathcal{O}_\infty$. So, the minimal numbers of bends to reach a destination orientation, referred to as $minBends(o)$ for $o \in \mathcal{O}_\infty$, are computed using Algorithm 8 with $\gamma(r) = 1$ for all $r \in \mathcal{R}_\infty$, as illustrated on Figure 5.2 with $\mathcal{O}^{dest.} = \{o_4\}$.

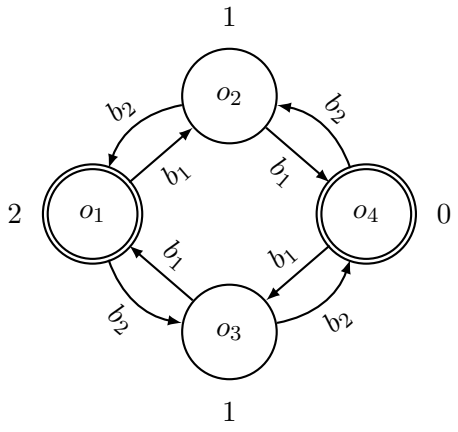


Figure 5.2 – Minimal number of bends to reach a destination orientation.

Algorithm 8: Evaluate the minimal costs $\gamma(o)$ to reach a destination orientation

Input:

- Kernel of reachable orientations: $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$
- Cost function: $\gamma(o) \qquad \forall o \in \mathcal{O}_\infty$

```
OpenList ← ∅
for o ∈ O^dest. do
    γ(o) ← 0
    Add node o in OpenList with value γ(o)
while OpenList ≠ ∅ do
    Remove o with the smallest value γ(o) from OpenList
    for r ∈ R_∞^inc.(o) do
        if o_r^- was not visited before then
            γ(o_r^-) ← ∞
        if γ(o_r^-) > γ(o_r^+) + γ(r) then
            γ(o_r^-) ← γ(o_r^+) + γ(r)
            Add node o_r^- in OpenList with value γ(o_r^-)
return γ(o) for o ∈ O_∞
```

5.1.3 Space of candidate orientations

The candidate orientations for each of the N_S possible segments of neutral fibre $\mathcal{F}_\pi$ are the reachable orientations which are on a path between the origin and destination orientations in kernel $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$. These paths should have a maximal length of $N_S - 1$ to satisfy the constraint on the maximum number of bends (see Section 4.2.1 on page 47).

Definition 17: Candidate orientation

Let k be a positive integer in $[\![1, N_S]\!]$. A *k-candidate orientation* is a reachable orientation $o \in \mathcal{O}_\infty$ which can be reached using exactly $k-1$ bends from catalogue $B_{cat.}$ and from which a destination orientation in $\mathcal{O}^{dest.}$ can be reached using at most $N_S - k$ bends from catalogue $B_{cat.}$, which can be written as:

$$\begin{cases} \exists (b_1, ..., b_{k-1}) \in B_{cat.}^{k-1} \;\mid\; o = rot_{M_{b_{k-1}}} \circ ... \circ rot_{M_{b_1}} \left(o^{ori.}\right) \\ minBends(o) \leq N_S - k \end{cases}$$

The set of k-candidate orientations is referred to as $\mathcal{O}_k$.

Definition 18: Candidate orientation change

Let k be a positive integer in $[\![1, N_S - 1]\!]$. A *k-candidate orientation change* is a reachable orientation change $r \in \mathcal{R}$ which verifies:

$$o_r^- \in \mathcal{O}_k \qquad\qquad o_r^+ \in \mathcal{O}_{k+1}$$

The set of k-candidate orientation changes is referred to as $\mathcal{R}_k$.

In the following sections, $\mathcal{R}_k^{inc.}(o)$ and $\mathcal{R}_k^{out.}(o)$ refer respectively to the incoming and outgoing candidate orientation changes into/from a candidate orientation $o \in \mathcal{O}_k$, respectively for $k \in [\![2, N_S]\!]$ and $k \in [\![1, N_S - 1]\!]$. The *space of candidate orientations* can be completely described by enumerating the k-candidate orientations $\mathcal{O}_k$, for $k \in [\![1, N_S]\!]$, and the k-candidate orientation changes $\mathcal{R}_k$, for $k \in [\![1, N_S - 1]\!]$. It can be represented by the directed acyclic graph $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$ shown on Figure 5.3.

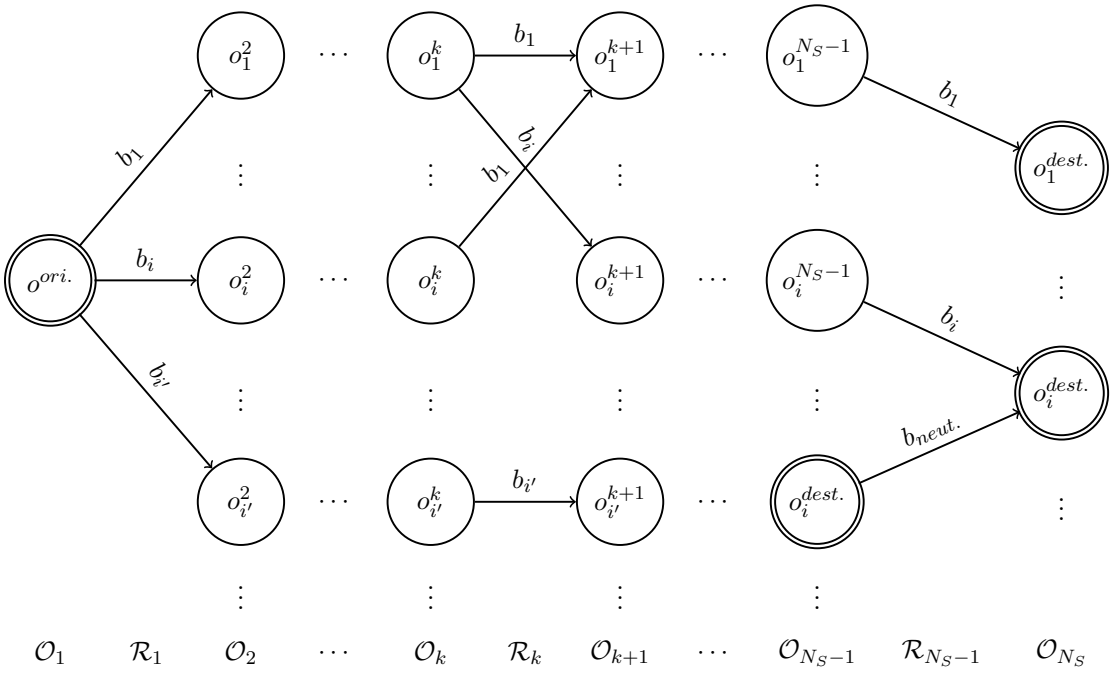


Figure 5.3 – Form of the space of candidate orientations $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$.

Algorithm 9 on the next page recursively generates $\mathcal{O}_{k+1}$ and $\mathcal{R}_{k+1}$ by applying the orientation changes of kernel $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$ from each k-candidate orientation of $\mathcal{O}_k$ only if the distance from the resulting orientation to a destination orientation is lower than the number of remaining bends. $\mathcal{O}_1$ is initialised with the origin orientation $o^{ori.}$. Furthermore, if a destination orientation $o^{dest.} \in \mathcal{O}^{dest.}$ is a k-candidate orientation and is contained in $\mathcal{O}_k$, a neutral bend which does not change the orientation is applied in order to maintain the orientation $o^{dest.}$ in the $k+1$-candidate orientations of $\mathcal{O}_{k+1}$. Using this trick, it is possible to solve a unique MILP model to get the optimal waveguide using up to $N_S - 1$ bends from catalogue $B_{cat.}$, rather than solving N_S MILP models to get the optimal solutions using exactly k bends, for $k \in [\![0, N_S - 1]\!]$.

Definition 19: Neutral bend

A *neutral bend* is a bend $b_{neut.} \in \mathcal{B}$ with a zero cost, which does not change the orientation and merges the length limitations of the previous and next segments. In other words, $b_{neut.}$ satisfies:

$$rot_{M_{b_{neut.}}} = Id \qquad \gamma_{b_{neut.}} = 0 \qquad L_{b_{neut.}} = -\frac{L^{min}}{2}$$

Figure 5.4 on the following page shows the candidate orientation space $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$ for the previous example with $N_S = 3$.

Algorithm 9: Generate the free candidate orientation space $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$

Input:

- Origin orientation: $o^{ori.}$
- Maximal number of segments: N_S
- Kernel of reachable orientations: $G\left(\mathcal{O}_\infty, \mathcal{R}_\infty\right)$
- Minimal number of bends to reach a destination orientation: $minBends\left(o\right) \qquad o \in \mathcal{O}_\infty$

$\mathcal{O}_1 \leftarrow \{o^{ori.}\}$
for $k \in [\![1, N_S - 1]\!]$ **do**
  $\mathcal{O}_{k+1} \leftarrow \varnothing$
  $\mathcal{R}_k \leftarrow \varnothing$
  for $o \in \mathcal{O}_k$ **do**
    for $r \in \mathcal{R}_\infty^{out.}\left(o\right)$ **do**
      if $minBends\left(o_r^+\right) \leq N_S - k$ **then**
        Add node o_r^+ in $\mathcal{O}_{k+1}$
        Add edge r in $\mathcal{R}_k$
    if $o \in \mathcal{O}^{dest.}$ **then**
      Add node o in $\mathcal{O}_{k+1}$
      Add edge $(o, o, b_{neut.})$ in $\mathcal{R}_k$
return $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$

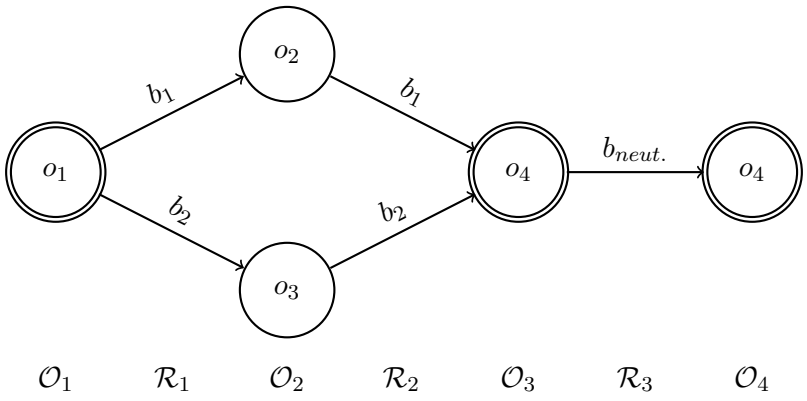


Figure 5.4 – Space of candidate orientations $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$ generated by Algorithm 9.

5.2 MILP formulation

The FWRP can be formulated as a decision problem which consists in choosing the bends of waveguide π among the bend catalogue $B_{cat.}$ and the neutral bend $b_{neut.}$, and defining the lengths of the straight sections between these bends. The cost γ_π of waveguide π must be as low as possible. In what follows, it is assumed that the reachable orientation space $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$

is not empty. This section presents the MILP model of the FWRP and details its criterion and its constraints.

5.2.1 Trivial case with $N_S = 1$

Model

When the maximum number of segments N_S is equal to 1, the waveguide contains only one straight section $u_{\pi,1}$ of length ℓ_1, as shown on Figure 5.5. The FWRP has a solution only if $o^{ori.} \in \mathcal{O}^{dest.}$ because no bend can be applied. If this statement is true, the FWRP can be solved with the following simple LP problem, referred to as LP_0, which minimises length ℓ_1 and places the origin p_1 and the destination p_2 of the neutral fibre $\mathcal{F}_\pi$:

$$\text{minimise } \mu\ell_1 \tag{5.1}$$
$$\text{subject to:}$$
$$p_1 \in \mathcal{P}^{ori.} \tag{5.2}$$
$$p_2 \in \mathcal{P}^{dest.} \tag{5.3}$$
$$\ell_1 \geq L^{min} \tag{5.4}$$
$$\overrightarrow{p_1 p_2} = \ell_1 \overrightarrow{e_{o^{ori.},z}} \tag{5.5}$$
$$\ell_1 \in \mathbb{R}^+ \tag{5.6}$$
$$p_1, p_2 \in \mathbb{R}^3 \tag{5.7}$$

Constraints 5.2 and 5.3 respectively state that neutral fibre $\mathcal{F}_\pi$ must start from the origin polyhedron and reach the destination polyhedron. Such an inclusion constraint within a convex polyhedron can be easily expressed as a set of linear constraints. Constraint 5.4 ensures that the length of the straight section satisfies the minimum length L^{min}. Constraint 5.5 defines the coordinates of the second point of the neutral fibre $\mathcal{F}_\pi$ from the coordinates of the first point, the origin orientation $o^{ori.}$, and the length of the segment (see Figure 5.5).

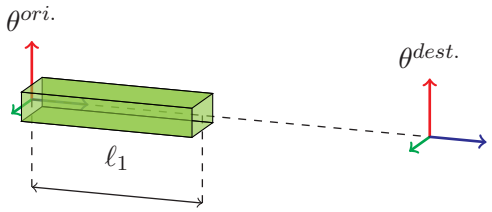


Figure 5.5 – Trivial case with $N_S = 1$.

Solutions

If (ℓ_1^*, p_1^*, p_2^*) is a solution of LP_0, then an optimal solution of the FWRP is the waveguide π^* defined by the origin configuration $\theta_{\pi^*}^{ori.} = (p_1^*, o^{ori.})$ and the composition $\sigma_{\pi^*} = u_{\ell_1^*}$. The destination configuration of this waveguide is then $\theta_{\pi^*}^{dest.} = (p_2^*, o^{ori.})$.

On the opposite, when LP_0 does not have a solution, the FWRP does not have a solution either.

5.2.2 General case with $N_S \geq 2$

Model

In the general case with $N_S \geq 2$, the FWRP cannot be written as a linear program. Indeed, integer variables must be introduced to choose the bends of waveguide π. So, the FWRP can be formulated as a MILP model that contains three kinds of variables:

- *bend variables* $x_{k,r}$ such that, for $k \in [\![1, N_S - 1]\!]$ and $r \in \mathcal{R}_k$, integer variable $x_{k,r}$ takes value 1 if orientation change r is applied at the end point of the k^{th} segment of the neutral fibre (that means the k^{th} and $k+1^{\text{th}}$ segments have orientations o_r^- and o_r^+ respectively), 0 otherwise;

- *length variables* ℓ_k such that, for $k \in [\![1, N_S]\!]$, real variable ℓ_k is the length of the k^{th} segment of the neutral fibre;

- *position variables* $p_k = (p_{k,x}, p_{k,y}, p_{k,z})$ such that, for $k \in [\![1, N_S + 1]\!]$, real variable $p_{k,x}$ (respectively $p_{k,y}$ and $p_{k,z}$) is the x-coordinate (respectively y-coordinate and z-coordinate) of the k^{th} point of the neutral fibre.

With these decision variables, a MILP formulation of the FWRP can be written as follows:

$$\text{minimise} \sum_{k=1}^{N_S-1} \sum_{r \in \mathcal{R}_k} \gamma_{b_r} x_{k,r} + \mu \sum_{k=1}^{N_S} \ell_k \tag{5.8}$$

subject to:

$$\sum_{r \in \mathcal{R}_k} x_{k,r} = 1 \qquad \forall k \in [\![1, N_S - 1]\!] \tag{5.9}$$

$$\sum_{r \in \mathcal{R}_k^{out.}(o)} x_{k+1,r} = \sum_{r \in \mathcal{R}_k^{inc.}(o)} x_{k,r} \qquad \forall k \in [\![1, N_S - 2]\!]\,,\ \forall o \in \mathcal{O}_k \tag{5.10}$$

$$\sum_{\substack{r \in \mathcal{R}_k \\ b_r = b_{neut.}}} x_{k,r} \leq \sum_{\substack{r \in \mathcal{R}_{k+1} \\ b_r = b_{neut.}}} x_{k+1,r} \qquad \forall k \in [\![1, N_S - 2]\!] \tag{5.11}$$

$$p_1 \in \mathcal{P}^{ori.} \tag{5.12}$$

$$p_{N_S+1} \in \mathcal{P}^{dest.} \tag{5.13}$$

$$\overrightarrow{p_1 p_2} = \ell_1 \overrightarrow{e_{o^{ori.},z}} \tag{5.14}$$

$$\ell_k \geq L^{min} + \sum_{r \in \mathcal{R}_{k-1}} L_{b_r} x_{k-1,r} + \sum_{r \in \mathcal{R}_k} L_{b_r} x_{k,r} \qquad \forall k \in [\![1, N_S]\!] \tag{5.15}$$

$$\overrightarrow{p_k p_{k+1}} \leq \ell_k \overrightarrow{e_{o_r^+,z}} + \overrightarrow{M_{succ.}}\,(1 - x_{k-1,r}) \qquad \forall k \in [\![2, N_S]\!]\,,\ \forall r \in \mathcal{R}_{k-1} \tag{5.16}$$

$$\overrightarrow{p_k p_{k+1}} \geq \ell_k \overrightarrow{e_{o_r^+,z}} - \overrightarrow{M_{succ.}}\,(1 - x_{k-1,r}) \qquad \forall k \in [\![2, N_S]\!]\,,\ \forall r \in \mathcal{R}_{k-1} \tag{5.17}$$

$$x_{k,r} \in \{0, 1\} \qquad \forall k \in [\![1, N_S - 1]\!]\,,\ \forall r \in \mathcal{R}_k \tag{5.18}$$

$$\ell_k \in \mathbb{R}^+ \qquad \forall k \in [\![1, N_S]\!] \tag{5.19}$$

$$p_k \in \mathbb{R}^3 \qquad \forall k \in [\![1, N_S + 1]\!] \tag{5.20}$$

Note that $\overrightarrow{v} \leq \overrightarrow{v'}$ (respectively $\overrightarrow{v} \geq \overrightarrow{v'}$) means $(v_x \leq v'_x) \wedge \left(v_y \leq v'_y\right) \wedge (v_z \leq v'_z)$ (respectively $(v_x \geq v'_x) \wedge \left(v_y \geq v'_y\right) \wedge (v_z \geq v'_z)$).

The criterion and constraints of this MILP model are detailed in the following sections. However, two parts can be identified in the MILP formulation of the FWRP:

- an *orientation sub-problem* (with Constraints 5.9-5.11) which corresponds to a multiple target shortest path problem in the space of candidate orientations $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$;

- a *3D-position sub-problem* (with Constraints 5.12-5.17) which defines restrictions on the vertices and lengths of the neutral fibre.

Both sub-problems are coupled through a set of *coupling constraints* composed of Constraints 5.15-5.17 that link position, length and bend variables.

Solutions

If $\left(\left(x^*_{k,r}\right)_{k\in[\![1,N_S-1]\!],\ r\in\mathcal{R}_k}, (\ell^*_k)_{k\in[\![1,N_S]\!]}, (p^*_k)_{k\in[\![1,N_S+1]\!]}\right)$ is a solution of the MILP model, then an optimal solution waveguide π^* of the FWRP can be rebuilt using Algorithm 10, after extracting the sequence of bends $(b^*_k)_{k\in[\![1,N_S-1]\!]}$. In particular, this algorithm subtracts the length contributions of the bends to each segment length in order to get the actual lengths of the straight sections (see Figure 5.6 on page 61).

Algorithm 10: Rebuild a waveguide π

Input:

- Origin point: $p^{ori.}$
- Bend sequence: $(b_k)_{k\in[\![1,N_\pi-1]\!]}$
- Segment lengths: $(\ell_k)_{k\in[\![1,N_\pi]\!]}$

$\theta^{ori.}_\pi \leftarrow (p^{ori.}, o^{ori.})$
$\sigma_\pi \leftarrow Id$
$L \leftarrow 0$
$L^- \leftarrow 0$
for $k \in [\![1, N_S - 1]\!]$ **do**
  $L \leftarrow L + \ell_k$
  if $b_k \neq b_{neut.}$ **then**
    $\sigma_\pi \leftarrow b_k \circ u_{L-L^- -L_{b_k}} \circ \sigma_\pi$
    $L^- \leftarrow L_{b_k}$
    $L \leftarrow 0$
$\sigma_\pi \leftarrow u_{L-L^-} \circ \sigma_\pi$
return $\pi = (\theta^{ori.}_\pi, \sigma_\pi)$

On the opposite, when the MILP model has no solution, then the FWRP does not have a solution either.

5.2.3 Orientation sub-problem

Definition

$$
\begin{aligned}
&\text{minimise} && \sum_{k=1}^{N_S-1} \sum_{r\in\mathcal{R}_k} \gamma_{b_r} x_{k,r} && \\
&\text{subject to:} && && \\
&&& \sum_{r\in\mathcal{R}_k} x_{k,r} = 1 && \forall k \in [\![1, N_S - 1]\!] \\
&\sum_{r\in\mathcal{R}^{out.}_k(o)} x_{k+1,r} = && \sum_{r\in\mathcal{R}^{inc.}_k(o)} x_{k,r} && \forall k \in [\![1, N_S - 2]\!],\ \forall o \in \mathcal{O}_k \\
&\sum_{\substack{r\in\mathcal{R}_k \\ b_r=b_{neut.}}} x_{k,r} \leq && \sum_{\substack{r\in\mathcal{R}_{k+1} \\ b_r=b_{neut.}}} x_{k+1,r} && \forall k \in [\![1, N_S - 2]\!] \\
&&& x_{k,r} \in \{0, 1\} && \forall k \in [\![1, N_S - 1]\!],\ \forall r \in \mathcal{R}_k
\end{aligned}
$$

The *orientation sub-problem* aims at finding a bend combination that allows waveguide π to reach a destination orientation in $\mathcal{O}^{dest.}$ from the origin orientation $o^{ori.}$. It is a multiple target

shortest path problem in the space of candidate orientations $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$. Constraints 5.9 ensure that exactly one orientation change is selected at the end of the k^{th} segment of the neutral fibre. As the last candidate orientation set $\mathcal{O}_{N_S}$ contains only destination orientations, these constraints also ensure that a destination orientation is reached. Then, like the natural linear programming formulation of the shortest path problem, Constraints 5.10 force each k-candidate orientation $o \in \mathcal{O}_k$ to have as many outgoing applied orientation changes as incoming ones, except for the origin and destination orientations.

Symmetry breaking for neutral bends

$$\sum_{\substack{r \in \mathcal{R}_k \\ b_r = b_{neut.}}} x_{k,r} \leq \sum_{\substack{r \in \mathcal{R}_{k+1} \\ b_r = b_{neut.}}} x_{k+1,r} \quad \forall k \in [\![1, N_S - 2]\!]$$

Due to the introduction of the neutral bend $b_{neut.}$, a solution path in the space of candidate orientations $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$ which contains this neutral bend can be translated into several sequences of bends. For instance, if sequence $[b_{neut.}, b_1, b_2]$ is a solution, then $[b_1, b_{neut.}, b_2]$ and $[b_1, b_2, b_{neut.}]$ are solutions too and have the same cost. Constraints 5.11 break this symmetry by limiting the neutral bends to be applied at the end. To do so, if a neutral bend is used at step $k \in [\![1, N_S - 2]\!]$, then a neutral bend must be applied at step $k+1$. By induction, it can be easily proved that a neutral bend must also be used at every following step of the bend sequence. In this case, only $[b_1, b_2, b_{neut.}]$ is a valid sequence of bends, which reduces the valid search space by avoiding multiple occurrences of the same waveguide solution. As a consequence, the optimality of a solution can be proved faster.

5.2.4 3D-position sub-problem and coupling constraints

The *3D-position sub-problem* aims at finding a consistent neutral fibre $\mathcal{F}_\pi$ that connects the origin polyhedron $\mathcal{P}^{ori.}$ to the destination polyhedron $\mathcal{P}^{dest.}$. The polyline must satisfy a minimal length constraint on each segment, and must be as short as possible.

Impact on the criterion

$$\text{minimise} \sum_{k=1}^{N_S-1} \sum_{r \in \mathcal{R}_k} \gamma_{b_r} x_{k,r} + \mu \sum_{k=1}^{N_S} \ell_k$$

From the shortest path problem in $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$, the FWRP adds the contribution $\mu \sum_{k=1}^{N_S} \ell_k$ to the criterion which corresponds to the linear cost of the waveguide.

Connectivity

$$p_1 \in \mathcal{P}^{ori.}$$
$$p_{N_S+1} \in \mathcal{P}^{dest.}$$

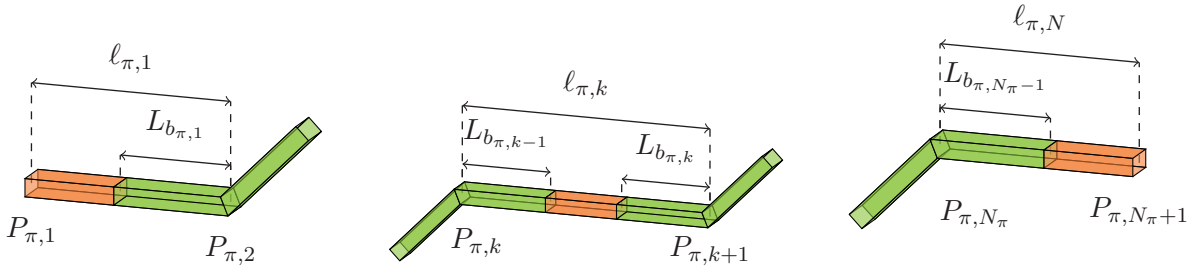
Constraints 5.12 and 5.13 respectively state that the neutral fibre must start from the origin polyhedron and reach the destination polyhedron.

Minimum length of straight sections

$$\ell_k \geq L^{min} + \sum_{r \in \mathcal{R}_{k-1}} L_{b_r} x_{k-1,r} + \sum_{r \in \mathcal{R}_k} L_{b_r} x_{k,r} \quad \forall k \in [\![1, N_S]\!]$$

Constraints 5.15 impose a minimal length on the segments. For the k^{th} segment, this minimum length is obtained from the minimal length L^{min} of straight sections and the respective contributions $\sum_{r \in \mathcal{R}_{k-1}} L_{b_r} x_{k-1,r}$ and $\sum_{r \in \mathcal{R}_k} L_{b_r} x_{k,r}$ of the previous and next bends (see Figure 5.6b). The first and last segments of neutral fibre $\mathcal{F}_\pi$ are particular cases, as shown on Figure 5.6a and Figure 5.6c (by convention, $\mathcal{R}_0 = \mathcal{R}_{N_S} = \varnothing$).

Moreover, when neutral bends are applied at the end, Constraints 5.15 and the neutral bend definition with $L_{b_{neut.}} = -\frac{L^{min}}{2}$ ensure that the minimum length constraint is applied on the total length of the segments around the neutral bends rather than on the length of each segment.



(a) First segment. (b) Intermediate segment. (c) Last segment.

Figure 5.6 – Portion of a waveguide between two successive break points of the neutral fibre $\mathcal{F}_\pi$ (the straight section with variable length is depicted in orange).

Property 1: Succession of neutral bends

Let $k \leq N_S - 1$ be a positive integer. If a neutral bend is applied at the end of the k'^{th} segment for $k' \in [\![k, N_S - 1]\!]$ and if the first neutral bend is applied at the end of the k^{th} segment, then the lower bound on the total length from the k^{th} segment to the last segment is defined by:

$$\sum_{k'=k}^{N_S} \ell_{k'} \geq L^{min} + \sum_{r \in \mathcal{R}_{k-1}} L_{b_r} x_{k-1,r}$$

Proof: *If neutral bends are applied at the end of the k'^{th} segment for $k' \in [\![k, N_S - 1]\!]$, then:*

$$\forall k' \in [\![k, N_S - 1]\!] \qquad \sum_{r \in \mathcal{R}_{k'}} L_{b_r} x_{k',r} = L_{b_{neut.}} = -\frac{L^{min}}{2}$$

So, Constraints 5.15 become:

- *for the k^{th} segment:*

$$\ell_k \geq \frac{L^{min}}{2} + \sum_{r \in \mathcal{R}_{k-1}} L_{b_r} x_{k-1,r}$$

- *for the k'^{th} segment with $k' \in [\![k+1, N_S - 1]\!]$:*

$$\ell_{k'} \geq 0$$

- *for the $N_S{}^{th}$ segment:*

$$\ell_{N_S} \geq \frac{L^{min}}{2}$$

Thus, the addition of Constraints 5.15 for $k' \in [\![k, N_S]\!]$ gives:

$$\sum_{k'=k}^{N_S} \ell_{k'} \geq L^{min} + \sum_{r \in \mathcal{R}_{k-1}} L_{b_r} x_{k-1,r}$$

Succession constraints

$$\begin{aligned}
\overrightarrow{p_1 p_2} &= \ell_1 \overrightarrow{e_{o^{ori.},z}} \\
\overrightarrow{p_k p_{k+1}} &\leq \ell_k \overrightarrow{e_{o_r^+,z}} + \overrightarrow{M_{succ.}}\,(1 - x_{k-1,r}) \quad \forall k \in [\![2, N_S]\!]\,,\ \forall r \in \mathcal{R}_{k-1} \\
\overrightarrow{p_k p_{k+1}} &\geq \ell_k \overrightarrow{e_{o_r^+,z}} - \overrightarrow{M_{succ.}}\,(1 - x_{k-1,r}) \quad \forall k \in [\![2, N_S]\!]\,,\ \forall r \in \mathcal{R}_{k-1}
\end{aligned}$$

Remind that the notation $\overrightarrow{v} \leq \overrightarrow{v'}$ (respectively $\overrightarrow{v} \geq \overrightarrow{v'}$) means $(v_x \leq v'_x) \wedge \left(v_y \leq v'_y\right) \wedge (v_z \leq v'_z)$ (respectively $(v_x \geq v'_x) \wedge \left(v_y \geq v'_y\right) \wedge (v_z \geq v'_z)$). Constraints 5.14, 5.16 and 5.17 define the coordinates of the $(k+1)^{\text{th}}$ point of neutral fibre $\mathcal{F}_\pi$ from the coordinates of the k^{th} point and the orientation and length of the k^{th} segment (see Figure 5.6). To do so, Constraints 5.16 and 5.17 use the big-M technique to linearise the constraints using the bend variables. If $x_{k-1,r} = 1$ for $k \in [\![2, N_S]\!]$ and $r \in \mathcal{R}_{k-1}$, then Constraints 5.16 and 5.17 give $\overrightarrow{p_k p_{k+1}} = \ell_k \overrightarrow{e_{o_r^+,z}}$, that means point p_{k+1} corresponds to point p_k translated along orientation $\overrightarrow{e_{o_r^+,z}}$ with length ℓ_k. On the opposite if $x_{k-1,r} = 0$, Constraints 5.16 and 5.17 give $\overrightarrow{p_k p_{k+1}} \leq \ell_k \overrightarrow{e_{o_r^+,z}} + \overrightarrow{M_{succ.}}$ and $\overrightarrow{p_k p_{k+1}} \geq \ell_k \overrightarrow{e_{o_r^+,z}} - \overrightarrow{M_{succ.}}$ respectively. Since these constraints must be disabled when $x_{k-1,r} = 0$, vector $\overrightarrow{M_{succ.}}$ must be chosen in a way that the position of point p_{k+1} is not limited in this case.

In practice, the instances of the FWRP aim to connect two components in a satellite with bounded dimensions. So there exists an upper bound L_{UB} such that for any index $k \in [\![1, N_S]\!]$:

$$\|\overrightarrow{p_k p_{k+1}}\| \leq L_{UB} \qquad\qquad \ell_k \leq L_{UB}$$

Typically, the maximal distance between two points of the satellite can be used as L_{UB} value.

Property 2: Big-M value in succession constraints

Let L_{UB} be an upper bound of the length of a segment for the considered FWRP instance. For $k \in [\![2, N_S]\!]$ and $r \in \mathcal{R}_{k-1}$, if $\overrightarrow{M_{succ.}} = (2L_{UB}, 2L_{UB}, 2L_{UB})$ then Constraints 5.16 and 5.17 are disabled when $x_{k-1,r} = 0$ in the sense that they do not constrain the position of point p_{k+1}.

Proof: *Let k be an index in $[\![2, N_S]\!]$ and r be a $(k-1)$-candidate orientation change $\mathcal{R}_{k-1}$ such that $x_{k-1,r} = 0$. Constraints 5.16 and 5.17 with $\overrightarrow{M_{succ.}} = (2L_{UB}, 2L_{UB}, 2L_{UB})$ give:*

$$\ell_k \overrightarrow{e_{o_r^+,z}} - (2L_{UB}, 2L_{UB}, 2L_{UB}) \leq \overrightarrow{p_k p_{k+1}} \leq \ell_k \overrightarrow{e_{o_r^+,z}} + (2L_{UB}, 2L_{UB}, 2L_{UB})$$

However, $\ell_k \leq L_{UB}$ by definition of the upper bound. It results that:

$$-(L_{UB}, L_{UB}, L_{UB}) \leq \ell_k \overrightarrow{e_{o_r^+,z}} \leq (L_{UB}, L_{UB}, L_{UB})$$

Finally:

$$-(L_{UB}, L_{UB}, L_{UB}) \leq \overrightarrow{p_k p_{k+1}} \leq (L_{UB}, L_{UB}, L_{UB})$$

These last equations are always satisfied since $\|\overrightarrow{p_k p_{k+1}}\| \leq L_{UB}$. *Q.E.D.*

5.3 Experiments on the FWRP

In this section, test instances of the FWRP derived from actual industrial cases are detailed (see Section 5.3.1) and used to experiment the MILP formulation. First, the dimensions of the kernel of reachable orientations $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$ and MILP models are respectively studied in Section 5.3.2 and Section 5.3.3. Then, Section 5.3.4 presents the performances of the MILP approach for the resolution of the test instances.

5.3.1 Instance sets

The MILP formulation as well as all the resolution approaches of the FWRP that will be introduced later have been experimented on three instance sets corresponding to different bend catalogues with the widely used gauge WR75. These catalogues are:

- catalogue $B^{90°}_{cat.}$ with 90°-bends around the four sides of the waveguide cross-section;
- catalogue $B^{45°}_{cat.}$ with 45° and 90° bends;
- catalogue $B^{30°}_{cat.}$ with 30°, 45°, 60°, and 90° bends.

All of them also contain a twist. So bend catalogues $B^{90°}_{cat.}$, $B^{45°}_{cat.}$ and $B^{30°}_{cat.}$ contain 5, 9 and 17 bends respectively. The features of these bends are detailed in Table 11 on page 170 in the appendices.

For all instance sets, the minimal length is $L^{min} = 5$, the linear cost is unit ($\mu = 1$) and each bend $b \in B_{cat.}$ of the catalogue has a cost $\gamma_b = 100$, except for the twist which is more expensive in pratice and has a cost $\gamma_b = 1000$. Remind that the cost γ_b of a bend $b \in B_{cat.}$ relatively to the linear cost μ can be interpreted as whether it is preferable to use bend b rather than none if the total length of the waveguide is reduced by at least $\frac{\gamma_b}{\mu}$ as explained in Section 4.2.2. Here, it is preferable to use an additional bend if it reduces the total length of 100 (or 1000 for a twist).

For each bend catalogue $B_{cat.}$, a set of 100 instances has been generated. To build an instance, the origin and destination orientations $o^{ori.}$ and $o^{dest.}$ are randomly drawn among the kernel of reachable orientations $\mathcal{O}_\infty$ of catalogue $B_{cat.}$. The origin and destination polyhedrons $\mathcal{P}^{ori.}$ and $\mathcal{P}^{dest.}$ are points generated inside a volume which corresponds to the real size of a satellite. The dimensions of this volume are $2750 \times 2650 \times 6250$ (in millimeters) as detailed in Table 6 on page 167. The generation of candidate points is performed using Bridson's algorithm [11], a maximum Poisson-disk sampling method, and then the origin and destination points are randomly selected among these candidates.

In the results presented in Part II, the instances for each bend catalogue are labelled with an ID number "#" ordered by cost of the optimal solution found with the MILP approach. The reader should note that instances with the same ID located in different instance sets are completely distinct since the orientations are drawn in different kernels of reachable orientations.

The purpose is to find a resolution method that ensures to solve all instances of the three sets, especially instances using bend catalogue $B^{30°}_{cat.}$ which corresponds to a typical catalogue used in an industrial context. The resolution of such instances will be requested thousands

of times during the design of the RF-harness. Indeed, a real satellite may contains thousands of waveguides and the routing of each waveguide can be performed several times by moving waypoints between each iteration. A waypoint is an intermediate polyhedron through which the waveguide must pass. As a consequence, a routing instance must be solved as fast as possible, ideally within a second.

5.3.2 Size of the kernel of reachable orientations

The first step to solve a routing instance is to enumerate the kernel of reachable orientations $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$ associated with bend catalogue $B_{cat.}$. As shown in Table 5.1, this enumeration is pretty fast, taking only a few seconds even for an industrial catalogue like $B^{30°}_{cat.}$. Remind that this operation only needs to be performed once per bend catalogue by considering relative orientations (see Section 5.1.1 on page 51).

	Catalogue $B^{90°}_{cat.}$			**Catalogue $B^{45°}_{cat.}$**			**Catalogue $B^{30°}_{cat.}$**		
N_S	$\lvert\mathcal{O}_\infty\rvert$	$\lvert\mathcal{R}_\infty\rvert$	**Runtime (in ms)**	$\lvert\mathcal{O}_\infty\rvert$	$\lvert\mathcal{R}_\infty\rvert$	**Runtime (in ms)**	$\lvert\mathcal{O}_\infty\rvert$	$\lvert\mathcal{R}_\infty\rvert$	**Runtime (in ms)**
0	1	0	<1	1	0	<1	1	0	<1
1	6	5	1	10	9	<1	18	17	<1
2	17	30	<1	37	74	1	89	210	5
3	24	85	1	94	285	5	382	1129	55
4	24	120	1	112	654	17	830	4350	563
5	24	120	1	112	752	23	1072	8606	1785
6	24	120	1	112	752	24	1104	10800	2723
7	24	120	1	112	752	26	1104	11088	2892
8	24	120	2	112	752	28	1104	11088	3052
9	24	120	2	112	752	29	1104	11088	3199
10	24	120	1	112	752	36	1104	11088	3176

Table 5.1 – Size of the kernel of reachable orientations $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$.

Furthermore, it can be noticed that the number of reachable orientations $|\mathcal{O}_\infty|$ if finite and reaches a maximal threshold when the maximal number of segments N_S increases, as illustrated on Figure 5.7. This is due to the global attachability constraints which restrict the possible orientations $\mathcal{O}^{int.}$ to a finite set.

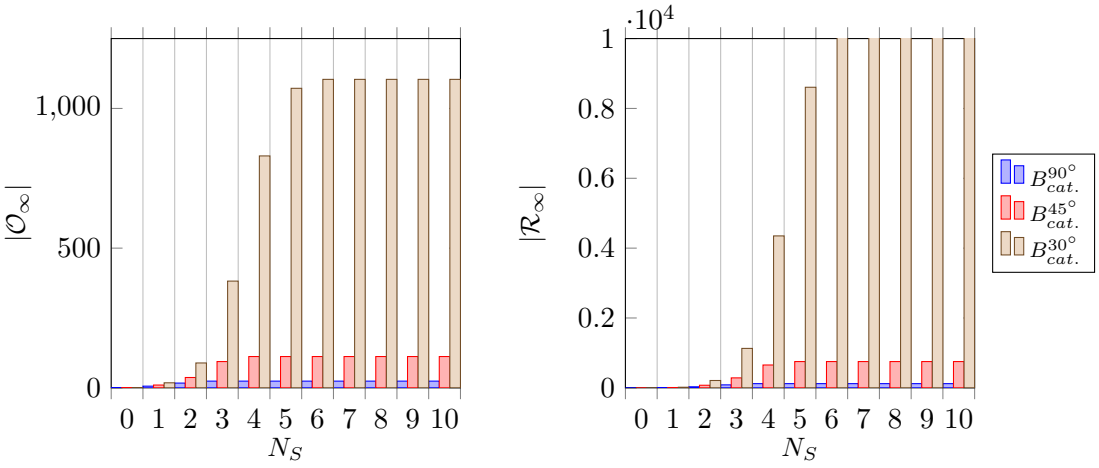


Figure 5.7 – Threshold on the size of the kernel of reachable orientations $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$.

5.3.3 MILP model sizes

Figure 5.8 presents the MILP model sizes for the test instances in terms of number of variables and constraints. It appears that the total number of variables (integer and real) remains reasonably low although it grows with the number of bends in catalogue $B_{cat.}$. Remind that the number of integer variables corresponds to the number of candidate orientation changes in $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$.

	Catalogue $B_{cat.}^{90°}$			Catalogue $B_{cat.}^{45°}$			Catalogue $B_{cat.}^{30°}$		
	Min.	**Mean**	**Max.**	**Min.**	**Mean**	**Max.**	**Min.**	**Mean**	**Max.**
# variables	207	**278.84**	341	652	**851.66**	1072	254	**1457.36**	3549
# constraints	1136	**1582.08**	1968	3921	**5150.5**	6509	1457	**8924.19**	21826

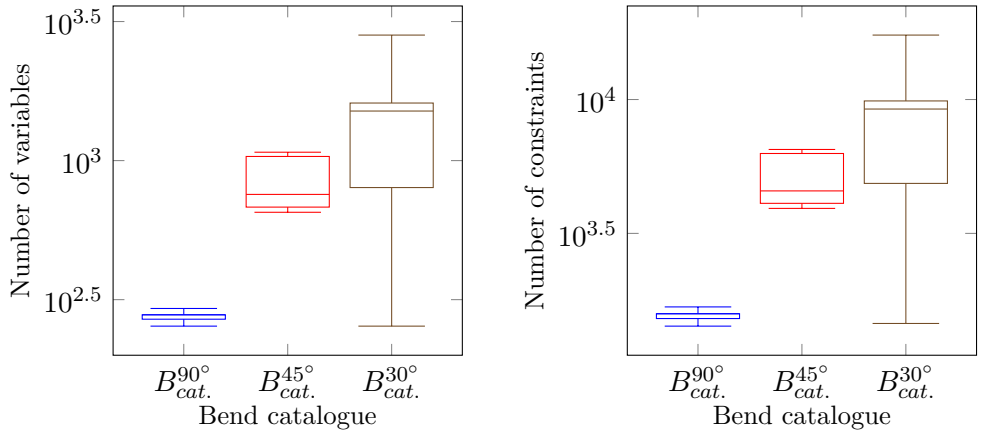


Figure 5.8 – Size of the FWRP MILP models.

5.3.4 Test instances resolution

All experiments presented in this thesis have been performed using a single thread on an Intel® Core i5-6500 CPU 3.20 GHz processor with 23.4 GB of RAM. The MILP formulation has been implemented in Java using the Google OR-Tools library (version 9.1.9490) [73] and instances have been solved using the SCIP solver (version 7.0.1) [32, 33]. In the results presented in this section, the maximum number of segments is $N_S = 7$.

Two kinds of resolutions are compared: a resolution until the optimal solution is reached depending on the CPU time allowed for each instance and another one stopped as soon as a first solution is found. Figure 5.10 shows the evolution of the percentage of instances solved over time, for each catalogue. It clearly appears that the optimal resolution requires an excessive runtime: even with the smallest bend catalogue $B_{cat.}^{90°}$, more than 10 seconds are needed to solve the most difficult instances. The first solutions are found within a second for bend catalogue $B_{cat.}^{90°}$ only, and less than 50% of the instances have been solved after 10 seconds for the industrial catalogue $B_{cat.}^{30°}$. Furthermore, the risk to obtain a first solution with a poor quality raises quickly with the size of the bend catalogue (see Figure 5.9). For around

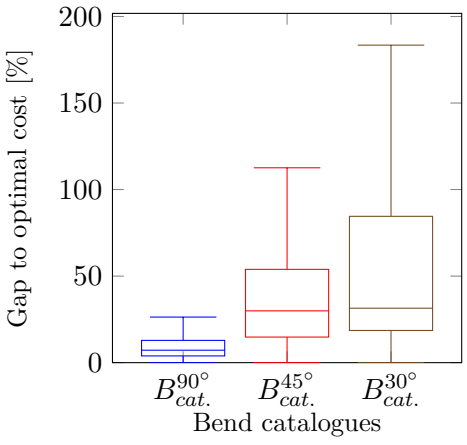


Figure 5.9 – Gaps to the optimal cost per bend catalogue $B_{cat.}$ for the first solution found.

20% of the first solutions found for catalogue $B_{cat.}^{30°}$, the cost is more than twice the cost of an optimal solution.

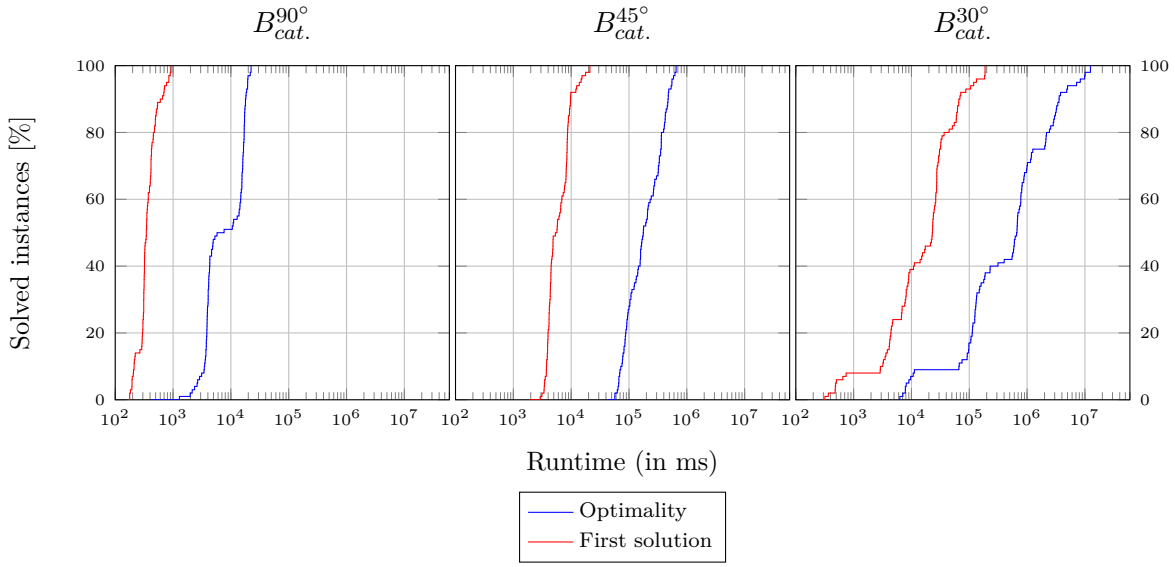


Figure 5.10 – Evolution of the success rates with MILP for the FWRP.

Table 5.2 on the facing page, Table 5.3 on page 68 and Table 5.4 on page 69 present the results respectively obtained on the $B_{cat.}^{90°}$, $B_{cat.}^{45°}$ and $B_{cat.}^{30°}$ instances. All result tables introduced in this thesis detail for each instance the cost, linear cost and bend cost of the solutions found, the gap to the optimal solution, the runtime, and the number of iterations performed. When the solution is optimal, the details of the result are printed in bold red.

It appears that the solutions found by the MILP formulation are quite realistic for a waveguide designer, as shown on Figure 5.11. Nevertheless, if this approach seems to be attractive because it guarantees to find optimal solutions, it can only be used in pratice for solving the FWRP with small catalogues. In order to be able to make a large number of iterations during the RF-harness design, a faster and more efficient approach is required.

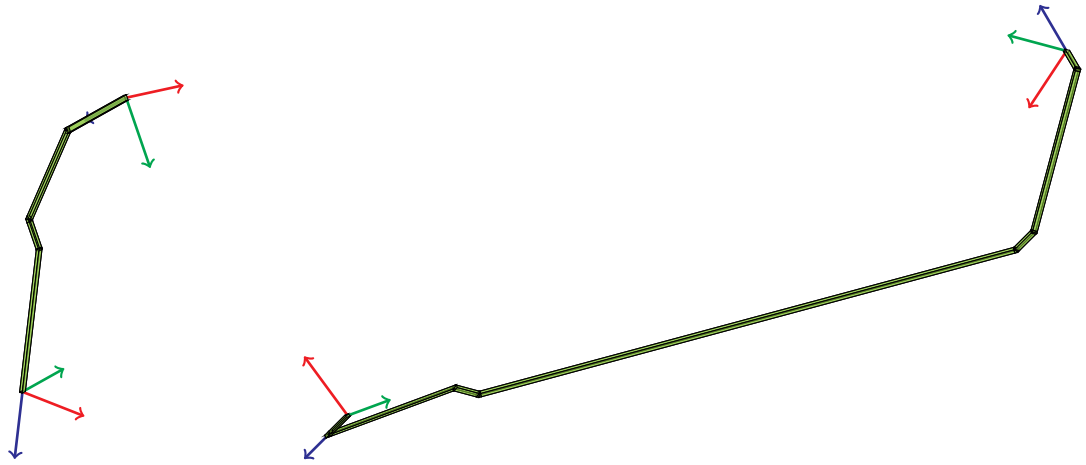


Figure 5.11 – Examples of waveguides routed using MILP for the FWRP.

Inst.		MILP optimal						MILP first					
#	Opti. cost	Cost	Gap [%]	Linear cost	Bend cost	Runtime (ms)	Iter.	Cost	Gap [%]	Linear cost	Bend cost	Runtime (ms)	Iter.
1	625.4	**625.4**	**0**	**310.12**	**300**	**413**	**424**	965.4	54.365	435.23	500	287	25
2	700.2	**700.2**	**0**	**280.5**	**400**	**1288**	**2837**	2150.2	207.083	620.03	1500	307	5
3	771.85	**771.85**	**0**	**352.15**	**400**	**2604**	**2457**	944.25	22.336	524.55	400	415	459
4	871.89	**871.89**	**0**	**554.6**	**300**	**1978**	**2479**	932.49	6.95	615.2	300	174	7
5	930.71	**930.71**	**0**	**509**	**400**	**5767**	**4854**	1200.71	29.01	566.12	600	415	13
6	1220.12	**1220.12**	**0**	**1009.26**	**200**	**7637**	**6769**	2580.72	111.513	1050.55	1500	311	23
7	1361.62	**1361.62**	**0**	**835.48**	**500**	**2142**	**1675**	2471.62	81.521	1051.92	1400	657	722
8	1598.38	**1598.38**	**0**	**1072.24**	**500**	**2358**	**2212**	1658.98	3.791	1132.84	500	498	657
9	1600.82	**1600.82**	**0**	**1076.7**	**500**	**2664**	**2834**	1680.22	4.96	1150.05	500	350	24
10	1824.45	**1824.45**	**0**	**1507.15**	**300**	**3121**	**3015**	2155.05	18.121	1624.87	500	373	5
11	1844.96	**1844.96**	**0**	**1527.66**	**300**	**3589**	**3856**	2175.56	17.919	1645.39	500	194	27
12	1873.38	**1873.38**	**0**	**1556.09**	**300**	**4266**	**3889**	2115.18	12.907	1589.05	500	405	462
13	1930.45	**1930.45**	**0**	**1510.75**	**400**	**13852**	**13870**	2278.58	18.034	1641.97	600	320	16
14	1955.11	**1955.11**	**0**	**1637.81**	**300**	**3842**	**3399**	2285.71	16.91	1755.53	500	329	8
15	1998.61	**1998.61**	**0**	**1474.49**	**500**	**4332**	**4188**	2078.01	3.973	1547.84	500	343	26
16	2044.85	**2044.85**	**0**	**1623.13**	**400**	**10583**	**8210**	2305.45	12.744	1670.86	600	480	15
17	2093.77	**2093.77**	**0**	**1672.05**	**400**	**14399**	**14091**	2484.97	18.684	1850.37	600	316	21
18	2162.45	**2162.45**	**0**	**1636.31**	**500**	**3759**	**3763**	2241.85	3.672	1711.67	500	343	7
19	2460.45	**2460.45**	**0**	**2143.15**	**300**	**3430**	**3388**	2791.05	13.437	2260.88	500	207	30
20	2460.68	**2460.68**	**0**	**2038.96**	**400**	**13933**	**10723**	2721.28	10.591	2086.69	600	511	21
21	2502.23	**2502.23**	**0**	**2184.94**	**300**	**3477**	**3010**	2832.83	13.212	2302.66	500	197	25
22	2609.89	**2609.89**	**0**	**2399.03**	**200**	**16945**	**14602**	**2609.89**	**0**	**2399.03**	**200**	**293**	**9**
23	2635.01	**2635.01**	**0**	**2215.31**	**400**	**10953**	**9268**	2946.81	11.833	2318.27	600	691	825
24	2707.36	**2707.36**	**0**	**2392.08**	**300**	**3861**	**3399**	2986.76	10.32	2456.59	500	222	23
25	2717.92	**2717.92**	**0**	**2400.62**	**300**	**4642**	**3336**	2778.52	2.23	2461.22	300	174	5
26	2770.42	**2770.42**	**0**	**2350.72**	**400**	**15477**	**13699**	3049.82	10.085	2413.21	600	325	13
27	2781.24	**2781.24**	**0**	**2359.52**	**400**	**15551**	**15713**	3153.64	13.39	2523.08	600	714	670
28	2813.56	**2813.56**	**0**	**2498.28**	**300**	**4322**	**3853**	3153.56	12.084	2623.39	500	349	42
29	2817.31	**2817.31**	**0**	**2291.17**	**500**	**3883**	**3454**	2931.71	4.061	2401.53	500	428	221
30	2825.46	**2825.46**	**0**	**2614.6**	**200**	**19446**	**17193**	**2825.46**	**0**	**2614.6**	**200**	**299**	**9**
31	2835.45	**2835.45**	**0**	**2520.17**	**300**	**4054**	**3583**	3105.45	9.522	2577.3	500	392	15
32	2890.9	**2890.9**	**0**	**2366.78**	**500**	**3642**	**3758**	2970.3	2.747	2440.13	500	360	27
33	2910.48	**2910.48**	**0**	**2595.2**	**300**	**4828**	**4046**	5810.48	99.64	2601.64	3200	302	18
34	3001.61	**3001.61**	**0**	**2684.31**	**300**	**4635**	**3964**	3341.61	11.327	2811.43	500	338	9
35	3022.75	**3022.75**	**0**	**2496.61**	**500**	**3895**	**3507**	3092.75	2.316	2566.61	500	423	401
36	3053.38	**3053.38**	**0**	**2842.52**	**200**	**19553**	**17157**	3663.38	19.978	3026.77	600	299	29
37	3224.51	**3224.51**	**0**	**2802.8**	**400**	**15902**	**17260**	3536.31	9.67	2905.76	600	895	748
38	3290.53	**3290.53**	**0**	**2764.39**	**500**	**3767**	**3435**	3369.93	2.413	2841.77	500	403	23
39	3328.86	**3328.86**	**0**	**2907.14**	**400**	**14744**	**15173**	3570.66	7.264	2940.1	600	773	737
40	3348.99	**3348.99**	**0**	**2927.27**	**400**	**17001**	**14960**	3609.59	7.781	2974.99	600	476	19
41	3423.75	**3423.75**	**0**	**3212.89**	**200**	**17875**	**16246**	**3423.75**	**0**	**3212.89**	**200**	**296**	**9**
42	3432.64	**3432.64**	**0**	**3117.36**	**300**	**3752**	**3660**	3712.04	8.14	3181.86	500	214	24
43	3500.84	**3500.84**	**0**	**2976.72**	**500**	**4050**	**3709**	3580.24	2.268	3052.09	500	390	46
44	3505.7	**3505.7**	**0**	**3083.98**	**400**	**16545**	**13510**	3836.3	9.43	3201.71	600	455	12
45	3520.14	**3520.14**	**0**	**3204.86**	**300**	**3916**	**3314**	3860.14	9.659	3331.99	500	371	5
46	3665.8	**3665.8**	**0**	**3244.08**	**400**	**16523**	**15735**	3926.4	7.109	3291.81	600	418	18
47	3803.65	**3803.65**	**0**	**3383.95**	**400**	**15605**	**18058**	4124.85	8.445	3492.27	600	602	695
48	3815.88	**3815.88**	**0**	**3605.02**	**200**	**21624**	**19343**	**3815.88**	**0**	**3605.02**	**200**	**292**	**9**
49	3823.5	**3823.5**	**0**	**3399.76**	**400**	**15564**	**15668**	4154.1	8.647	3517.49	600	409	11
50	3829.26	**3829.26**	**0**	**3511.96**	**300**	**4203**	**4010**	5129.86	33.965	3710.16	1400	457	566
51	3831.5	**3831.5**	**0**	**3411.8**	**400**	**14317**	**15369**	4110.9	7.292	3474.29	600	315	12
52	3873.21	**3873.21**	**0**	**3662.35**	**200**	**18766**	**17660**	4413.21	13.942	3776.6	600	306	28
53	3908.18	**3908.18**	**0**	**3590.88**	**300**	**4287**	**3700**	4238.78	8.459	3708.6	500	195	40
54	3936.43	**3936.43**	**0**	**3412.3**	**500**	**2875**	**2901**	4006.43	1.778	3478.27	500	357	21
55	3976.19	**3976.19**	**0**	**3554.47**	**400**	**17407**	**15574**	4297.39	8.078	3662.79	600	535	16
56	4026.08	**4026.08**	**0**	**3604.36**	**400**	**17593**	**18825**	4356.68	8.211	3722.08	600	416	13
57	4087.37	**4087.37**	**0**	**3665.65**	**400**	**12855**	**12270**	4389.77	7.398	3759.21	600	846	937
58	4164.02	**4164.02**	**0**	**3744.32**	**400**	**16107**	**14194**	**4164.02**	**0**	**3744.32**	**400**	**413**	**538**
59	4276.56	**4276.56**	**0**	**3959.26**	**300**	**4894**	**3956**	4337.16	1.417	4019.86	300	183	5
60	4290.81	**4290.81**	**0**	**3975.53**	**300**	**4019**	**3262**	4630.81	7.924	4102.65	500	371	8
61	4303.24	**4303.24**	**0**	**3985.94**	**300**	**4950**	**4017**	4643.24	7.901	4113.06	500	345	8
62	4342.91	**4342.91**	**0**	**3921.19**	**400**	**16645**	**12915**	4622.31	6.433	3985.7	600	323	15
63	4360.45	**4360.45**	**0**	**4045.17**	**300**	**4135**	**3724**	6360.45	45.867	4045.17	2300	349	9
64	4390.66	**4390.66**	**0**	**3970.96**	**400**	**17897**	**19747**	4660.66	6.149	4026.06	600	322	15
65	4395.12	**4395.12**	**0**	**4079.84**	**300**	**4124**	**3464**	6395.12	45.505	4079.84	2300	205	6
66	4412.43	**4412.43**	**0**	**4201.57**	**200**	**22365**	**21485**	**4412.43**	**0**	**4201.57**	**200**	**290**	**10**
67	4415.24	**4415.24**	**0**	**3993.52**	**400**	**14833**	**16360**	4666.44	5.689	4035.88	600	922	1023
68	4443.45	**4443.45**	**0**	**4021.74**	**400**	**17239**	**17187**	4774.05	7.44	4139.46	600	415	15
69	4445.35	**4445.35**	**0**	**4021.62**	**400**	**14895**	**14766**	4684.13	5.371	4047.52	600	301	20
70	4567.38	**4567.38**	**0**	**4041.24**	**500**	**3927**	**3562**	4637.38	1.533	4109.22	500	320	22
71	4569.85	**4569.85**	**0**	**4150.15**	**400**	**17013**	**16513**	4909.85	7.44	4273.24	600	307	11
72	4611.59	**4611.59**	**0**	**4085.45**	**500**	**3642**	**3727**	4681.59	1.518	4153.44	500	317	21
73	4686.76	**4686.76**	**0**	**4265.04**	**400**	**18165**	**16191**	5007.96	6.853	4373.37	600	541	22
74	4803.26	**4803.26**	**0**	**4487.98**	**300**	**4321**	**3931**	6803.26	41.638	4487.98	2300	307	17
75	4861.29	**4861.29**	**0**	**4441.59**	**400**	**17319**	**18328**	4870.69	0.193	4446.95	400	326	47
76	4953.41	**4953.41**	**0**	**4533.71**	**400**	**11157**	**9559**	**4953.41**	**0**	**4533.71**	**400**	**450**	**664**
77	5056.52	**5056.52**	**0**	**4530.38**	**500**	**4145**	**4154**	5126.52	1.384	4600.38	500	494	491
78	5064.14	**5064.14**	**0**	**4748.86**	**300**	**5290**	**4547**	7964.14	57.265	4755.3	3200	307	15
79	5129.39	**5129.39**	**0**	**4707.67**	**400**	**16952**	**14833**	5450.59	6.262	4815.99	600	480	28
80	5170.66	**5170.66**	**0**	**4853.37**	**300**	**3525**	**3210**	5501.26	6.394	4971.09	500	365	7
81	5207.62	**5207.62**	**0**	**4785.9**	**400**	**16164**	**13248**	5538.22	6.348	4903.62	600	418	15
82	5211.32	**5211.32**	**0**	**4687.19**	**500**	**3701**	**3402**	5281.32	1.343	4753.16	500	312	20
83	5248.57	**5248.57**	**0**	**4828.87**	**400**	**15153**	**16085**	5527.97	5.323	4893.38	600	353	14
84	5508.9	**5508.9**	**0**	**5193.62**	**300**	**3944**	**3743**	5839.5	6.001	5311.34	500	313	26
85	5564.77	**5564.77**	**0**	**5249.49**	**300**	**4050**	**3246**	5844.17	5.021	5314	500	212	27
86	5567.35	**5567.35**	**0**	**5145.63**	**400**	**16061**	**13493**	5958.55	7.027	5323.96	600	321	22
87	5573.71	**5573.71**	**0**	**5154.01**	**400**	**15464**	**18430**	5834.31	4.676	5205.76	600	701	716
88	5629.91	**5629.91**	**0**	**5210.21**	**400**	**19777**	**19048**	7079.91	25.755	5549.74	1500	350	9
89	5709.24	**5709.24**	**0**	**5287.53**	**400**	**17123**	**18597**	6100.44	6.852	5465.85	600	318	28
90	5945.13	**5945.13**	**0**	**5418.99**	**500**	**3875**	**3410**	6024.53	1.336	5496.37	500	267	32
91	6047.33	**6047.33**	**0**	**5732.05**	**300**	**3837**	**3994**	8467.33	40.018	6152.05	2300	194	18
92	6266.31	**6266.31**	**0**	**5844.6**	**400**	**16843**	**16343**	6536.31	4.309	5901.72	600	433	24
93	6571.93	**6571.93**	**0**	**6256.65**	**300**	**4156**	**3726**	6902.53	5.03	6374.37	500	207	18
94	6600.83	**6600.83**	**0**	**6179.11**	**400**	**16016**	**18098**	6912.63	4.724	6282.07	600	773	693
95	6641.3	**6641.3**	**0**	**6221.6**	**400**	**17248**	**17574**	6911.3	4.065	6274.69	600	311	24
96	6748.17	**6748.17**	**0**	**6326.45**	**400**	**19509**	**20686**	7008.77	3.862	6374.17	600	515	24
97	6814.87	**6814.87**	**0**	**6395.17**	**400**	**16327**	**16985**	7094.27	4.1	6457.66	600	318	16
98	6858.28	**6858.28**	**0**	**6540.98**	**300**	**4047**	**3652**	**6858.28**	**0**	**6540.98**	**300**	**178**	**2**
99	6961.13	**6961.13**	**0**	**6434.99**	**500**	**3806**	**3523**	6979.93	0.27	6451.78	500	219	50
100	7350.81	**7350.81**	**0**	**6931.11**	**400**	**18382**	**18083**	7742.01	5.322	7107.42	600	315	28

Table 5.2 – Results for the FWRP with MILP on $B^{90^\circ}_{cat.}$ instances.

Inst.		MILP optimal						MILP first					
#	Opti. cost	Cost	Gap [%]	Linear cost	Bend cost	Runtime (ms)	Iter.	Cost	Gap [%]	Linear cost	Bend cost	Runtime (ms)	Iter.
1	678.28	**678.28**	**0**	**268.15**	**400**	**64710**	**9910**	1040.21	53.359	405.61	600	6643	127
2	806.99	**806.99**	**0**	**405.01**	**400**	**43418**	**9404**	832.08	3.109	526.57	300	4001	17
3	875.61	**875.61**	**0**	**351.44**	**500**	**91701**	**10553**	**875.61**	**0**	**351.44**	**500**	**15366**	**1849**
4	878.25	**878.25**	**0**	**356.09**	**500**	**56932**	**16001**	1327.15	51.113	696.53	600	4580	41
5	896.51	**896.51**	**0**	**482.74**	**400**	**214148**	**25627**	**896.51**	**0**	**482.74**	**400**	**12062**	**1258**
6	1140.42	**1140.42**	**0**	**823.12**	**300**	**156223**	**19820**	**1140.42**	**0**	**823.12**	**300**	**8095**	**11**
7	1392.39	**1392.39**	**0**	**984.42**	**400**	**62766**	**11310**	3705.67	166.138	1289.95	2400	3723	14
8	1403	**1403**	**0**	**989.24**	**400**	**178713**	**21970**	1709.06	21.815	1084.03	600	8696	41
9	1434.38	**1434.38**	**0**	**813.59**	**600**	**57100**	**9142**	2799.32	95.159	2188.5	600	4890	1279
10	1512.12	**1512.12**	**0**	**1099.97**	**400**	**199716**	**24434**	1798.01	18.907	1480.72	300	7591	12
11	1518.19	**1518.19**	**0**	**1106.24**	**400**	**66955**	**8603**	4375.34	188.194	1069.83	3300	3121	73
12	1519.8	**1519.8**	**0**	**906.76**	**600**	**330334**	**29471**	1917.62	26.176	1293	600	8498	56
13	1593.18	**1593.18**	**0**	**1281.87**	**300**	**85003**	**11349**	2005.58	25.886	1379	600	4493	112
14	1702.25	**1702.25**	**0**	**1092.99**	**600**	**103847**	**12757**	2159.46	26.859	1542.25	600	4192	19
15	1702.82	**1702.82**	**0**	**1188.41**	**500**	**124005**	**17981**	2081.9	22.262	1451.08	600	4387	65
16	1739.47	**1739.47**	**0**	**1431.95**	**300**	**97108**	**13699**	**1739.47**	**0**	**1431.95**	**300**	**4280**	**22**
17	1924.18	**1924.18**	**0**	**1510.42**	**400**	**279353**	**44884**	2650.46	37.745	2015.87	600	9850	22
18	2046.45	**2046.45**	**0**	**1526.1**	**500**	**335053**	**39318**	3155.04	54.172	1644.67	1500	9357	213
19	2075.62	**2075.62**	**0**	**1566.8**	**500**	**107862**	**13839**	2374.97	14.422	2067.44	300	3448	19
20	2102.73	**2102.73**	**0**	**1486.12**	**600**	**132109**	**20716**	2201.49	4.697	1572.89	600	17750	2711
21	2209.59	**2209.59**	**0**	**1598.37**	**600**	**72060**	**10998**	2684.62	21.499	2059.8	600	3971	22
22	2252.71	**2252.71**	**0**	**1746.11**	**500**	**317640**	**37351**	2547.09	13.068	2030.52	500	8988	50
23	2283.54	**2283.54**	**0**	**1774.92**	**500**	**553678**	**80581**	4051.82	77.436	2519.62	1500	8322	137
24	2306.69	**2306.69**	**0**	**1888.75**	**400**	**142333**	**21511**	3827.59	65.934	2305.22	1500	4771	39
25	2324.48	**2324.48**	**0**	**1707.27**	**600**	**219804**	**23230**	2692.55	15.835	2061.94	600	8437	52
26	2328.74	**2328.74**	**0**	**1822.15**	**500**	**406831**	**44185**	3265.26	40.215	2640.23	600	8753	20
27	2348.52	**2348.52**	**0**	**1733.47**	**600**	**66913**	**13305**	4110.03	75.005	2589.63	1500	3871	27
28	2400.84	**2400.84**	**0**	**1894.24**	**500**	**362514**	**52570**	3420.69	42.479	2786.09	600	5673	23
29	2427	**2427**	**0**	**1916.42**	**500**	**112429**	**18402**	3158.44	30.138	2527.82	600	5333	24
30	2460.66	**2460.66**	**0**	**2149.36**	**300**	**208968**	**30795**	2939.79	19.472	2309.18	600	4145	62
31	2461.95	**2461.95**	**0**	**2055.99**	**400**	**176162**	**32005**	2707.9	9.99	2091.09	600	4463	212
32	2482.31	**2482.31**	**0**	**1871.29**	**600**	**322726**	**39235**	**2482.31**	**0**	**1871.29**	**600**	**6919**	**436**
33	2532.98	**2532.98**	**0**	**2016.41**	**500**	**352336**	**50522**	3573.68	41.086	3053.48	500	9513	518
34	2558.32	**2558.32**	**0**	**2051.72**	**500**	**81876**	**13024**	4888.54	91.084	3368.14	1500	4352	33
35	2739.38	**2739.38**	**0**	**2236.56**	**500**	**165146**	**19935**	2899.61	5.849	2588.1	300	4666	7
36	2757.59	**2757.59**	**0**	**2254.77**	**500**	**141241**	**16016**	3318.48	20.34	3006.97	300	3714	9
37	2790.94	**2790.94**	**0**	**2274.37**	**500**	**85395**	**14908**	**2790.94**	**0**	**2274.37**	**500**	**6880**	**1971**
38	2800.14	**2800.14**	**0**	**2185.09**	**600**	**353331**	**43665**	3738.7	33.518	3112.06	600	8948	13
39	2800.54	**2800.54**	**0**	**2287.95**	**500**	**364335**	**32860**	3989.79	42.465	3374.59	600	8603	44
40	2824.52	**2824.52**	**0**	**2201.51**	**600**	**69686**	**12692**	3492.67	23.655	2867.84	600	13825	3370
41	2844.51	**2844.51**	**0**	**2223.72**	**600**	**75803**	**14593**	5309.02	86.641	4682.38	600	4144	24
42	2846.42	**2846.42**	**0**	**2338.01**	**500**	**94163**	**14122**	3211.42	12.823	2799.48	400	1907	770
43	2864.31	**2864.31**	**0**	**2249.06**	**600**	**77016**	**13242**	3216.8	12.306	2601.55	600	3947	43
44	2870.84	**2870.84**	**0**	**2458.69**	**400**	**397177**	**53405**	5276.16	83.784	3743.97	1500	8389	39
45	2916.82	**2916.82**	**0**	**2303.78**	**600**	**477187**	**65974**	4812.22	64.982	3280.03	1500	7667	37
46	2984.84	**2984.84**	**0**	**2677.51**	**300**	**589316**	**95773**	3748.14	25.573	3537.28	200	8641	24
47	3003.96	**3003.96**	**0**	**2387.15**	**600**	**277073**	**35346**	4430.96	47.504	3809.91	600	6333	83
48	3044.59	**3044.59**	**0**	**2532**	**500**	**263056**	**28215**	3679.19	20.843	3048.57	600	6148	70
49	3048.02	**3048.02**	**0**	**2437**	**600**	**424038**	**52708**	3596.75	18.003	2971.92	600	9696	82
50	3062.17	**3062.17**	**0**	**2555.57**	**500**	**361948**	**64683**	3994.28	30.439	3359.69	600	5783	31
51	3063.97	**3063.97**	**0**	**2450.93**	**600**	**108502**	**14708**	5534.39	80.628	3124.26	2400	3389	66
52	3145.64	**3145.64**	**0**	**2530.65**	**600**	**321551**	**30505**	3886.14	23.54	3253.51	600	12434	10
53	3151.84	**3151.84**	**0**	**2649.01**	**500**	**194627**	**33190**	3471.39	10.139	3159.88	300	4491	19
54	3152.42	**3152.42**	**0**	**2738.45**	**400**	**86276**	**14876**	6972.49	121.179	6347.67	600	4405	65
55	3244.19	**3244.19**	**0**	**2623.35**	**600**	**103476**	**17838**	4259.2	31.287	3630.4	600	5733	144
56	3255.09	**3255.09**	**0**	**2640.04**	**600**	**82160**	**12181**	3655.07	12.288	3028.43	600	4104	15
57	3285.07	**3285.07**	**0**	**2879.12**	**400**	**242683**	**27503**	4665.1	42.009	4050.31	600	6573	56
58	3299.84	**3299.84**	**0**	**2787.45**	**500**	**177472**	**30129**	4399.3	33.319	2875.13	1500	4793	72
59	3314.97	**3314.97**	**0**	**2705.7**	**600**	**158824**	**23082**	4840.78	46.028	3327.98	1500	3566	17
60	3342.01	**3342.01**	**0**	**2934.04**	**400**	**411055**	**44603**	4334.67	29.703	3715.84	600	8187	18
61	3351.18	**3351.18**	**0**	**2738.35**	**600**	**99953**	**14950**	4513.68	34.689	3887.04	600	3420	22
62	3503	**3503**	**0**	**2986.84**	**500**	**551390**	**80840**	4348.94	24.149	3712.33	600	8557	65
63	3533.16	**3533.16**	**0**	**3127.21**	**400**	**122305**	**21411**	10582.32	199.514	8166.4	2400	4853	55
64	3559.74	**3559.74**	**0**	**3041.56**	**500**	**361901**	**40103**	4795.89	34.726	3277.3	1500	9331	29
65	3584.91	**3584.91**	**0**	**3072.32**	**500**	**458204**	**43902**	4571.63	27.524	3952.6	600	14878	1514
66	3609.55	**3609.55**	**0**	**3203.4**	**400**	**232496**	**35689**	8243.92	128.392	7636.48	600	3981	49
67	3612.64	**3612.64**	**0**	**3003.59**	**600**	**167348**	**31178**	4187.36	15.908	3556.74	600	3482	17
68	3706.73	**3706.73**	**0**	**3097.47**	**600**	**158328**	**22456**	6441.47	73.778	4023.53	2400	4458	14
69	3738.02	**3738.02**	**0**	**3129.17**	**600**	**361330**	**38033**	4255.74	13.85	3944.43	300	7283	5
70	3741.47	**3741.47**	**0**	**3124.66**	**600**	**425925**	**52989**	4613.88	23.317	3985.28	600	8242	71
71	3760.72	**3760.72**	**0**	**3254.12**	**500**	**421107**	**55848**	4108.11	9.237	3790.81	300	8523	24
72	3763.18	**3763.18**	**0**	**3254.56**	**500**	**87461**	**13118**	6081.53	61.606	5454.9	600	3921	13
73	3784.22	**3784.22**	**0**	**3374.09**	**400**	**439386**	**54055**	6211.7	64.147	4695.13	1500	9732	158
74	3801.9	**3801.9**	**0**	**3295.31**	**500**	**652578**	**86666**	4931.14	29.702	4720.28	200	7849	7
75	3876.35	**3876.35**	**0**	**3367.74**	**500**	**90751**	**15853**	5824.47	50.256	4304.07	1500	3864	21
76	3930.82	**3930.82**	**0**	**3524.87**	**400**	**207750**	**31213**	4155.47	5.715	3536.64	600	3913	160
77	3982.73	**3982.73**	**0**	**3369.7**	**600**	**335019**	**56257**	5252.14	31.873	4617.54	600	6610	25
78	4104.72	**4104.72**	**0**	**3491.68**	**600**	**472649**	**65829**	6324.45	54.078	5798.31	500	21103	2616
79	4156.4	**4156.4**	**0**	**3549.16**	**600**	**133906**	**23075**	7184.4	72.852	4768.48	2400	4846	13
80	4172.59	**4172.59**	**0**	**3867.08**	**300**	**88513**	**13339**	**4172.59**	**0**	**3867.08**	**300**	**4249**	**21**
81	4209.75	**4209.75**	**0**	**3908.02**	**300**	**92240**	**13972**	4671.93	10.979	4041.32	600	4338	10
82	4320.79	**4320.79**	**0**	**3704.18**	**600**	**155768**	**25767**	4358.47	0.872	3731.88	600	4525	100
83	4421.04	**4421.04**	**0**	**3918.62**	**500**	**298463**	**30205**	5818.83	31.617	5190.03	600	5826	19
84	4463.73	**4463.73**	**0**	**3951.14**	**500**	**145531**	**25870**	7192.84	61.14	4775.1	2400	3776	42
85	4571.23	**4571.23**	**0**	**4064.64**	**500**	**467063**	**95034**	8131.86	77.892	5712.16	2400	2915	53
86	4639.61	**4639.61**	**0**	**4126.81**	**500**	**268985**	**27442**	4741.53	2.197	4133.88	600	8202	22
87	4715	**4715**	**0**	**4307.03**	**400**	**172226**	**23699**	8401.49	78.187	7784.68	600	4145	142
88	4923.06	**4923.06**	**0**	**4308.01**	**600**	**77918**	**13596**	9899.37	101.082	8378.97	1500	4190	20
89	5038.52	**5038.52**	**0**	**4531.93**	**500**	**624324**	**97728**	9449.02	87.535	8824.4	600	8139	70
90	5089.14	**5089.14**	**0**	**4582.54**	**500**	**79070**	**11595**	6104.66	19.955	5479.84	600	3740	17
91	5281.07	**5281.07**	**0**	**4668.44**	**600**	**535420**	**65532**	6548.98	24.008	5935.94	600	8522	56
92	5296.86	**5296.86**	**0**	**4782.45**	**500**	**158814**	**25621**	6693.65	26.37	6062.84	600	4451	33
93	5367.54	**5367.54**	**0**	**4752.28**	**600**	**66215**	**11698**	7949.02	48.094	7322.18	600	9743	1897
94	5565.08	**5565.08**	**0**	**5058.69**	**500**	**95428**	**13118**	7339.28	31.881	5822.91	1500	4150	79
95	5638.65	**5638.65**	**0**	**5232.7**	**400**	**210180**	**34236**	5899.5	4.626	5282.69	600	3905	100
96	5956.85	**5956.85**	**0**	**5347.99**	**600**	**487497**	**50561**	8676.65	45.658	8059.84	600	8444	25
97	6051.83	**6051.83**	**0**	**5535.46**	**500**	**69030**	**10826**	7519.27	24.248	5998.87	1500	4357	15
98	6159.55	**6159.55**	**0**	**5753.19**	**400**	**265469**	**41499**	6368.51	3.392	6057	300	3770	11
99	6513.65	**6513.65**	**0**	**5898.45**	**600**	**478801**	**51432**	9128.98	40.151	7618.6	1500	8854	233
100	6785.52	**6785.52**	**0**	**6168.71**	**600**	**161816**	**26820**	9016.24	32.875	8385.42	600	4859	90

Table 5.3 – Results for the FWRP with MILP on $B^{45^\circ}_{cat.}$ instances.

Inst.		MILP optimal						MILP first					
#	Opti. cost	Cost	Gap [%]	Linear cost	Bend cost	Runtime (ms)	Iter.	Cost	Gap [%]	Linear cost	Bend cost	Runtime (ms)	Iter.
1	775.36	**775.36**	**0**	**263.23**	**500**	**67016**	**11211**	788.21	1.658	276.08	500	8372	731
2	847.69	**847.69**	**0**	**434.25**	**400**	**401721**	**21244**	1662.1	96.073	1035.78	600	23839	46
3	883.95	**883.95**	**0**	**269.94**	**600**	**7112**	**4898**	905.44	2.431	292.2	600	503	38
4	948.57	**948.57**	**0**	**335.21**	**600**	**5992**	**3832**	1867.68	96.895	1251.98	600	3617	3030
5	1091.69	**1091.69**	**0**	**478.5**	**600**	**313999**	**15675**	1108.72	1.561	592.35	500	62482	3647
6	1245.49	**1245.49**	**0**	**740.53**	**500**	**2108995**	**67903**	1441.48	15.736	816.19	600	27440	326
7	1287.26	**1287.26**	**0**	**777.99**	**500**	**9708**	**5459**	1377.86	7.038	773.74	600	651	116
8	1378.31	**1378.31**	**0**	**872.95**	**500**	**1179922**	**159347**	1969.32	42.879	1342.01	600	15008	18
9	1444.21	**1444.21**	**0**	**1040.25**	**400**	**589424**	**27677**	1645.09	13.909	1337.95	300	27399	18
10	1551.37	**1551.37**	**0**	**1045.35**	**500**	**603899**	**25262**	2121.54	36.753	1495.09	600	30410	18
11	1556.78	**1556.78**	**0**	**1147.44**	**400**	**657762**	**29259**	1697.43	9.035	1389.7	300	14137	21
12	1597.2	**1597.2**	**0**	**981.95**	**600**	**10718**	**4519**	9586.15	500.183	8971.48	600	365	23
13	1608.93	**1608.93**	**0**	**1100.18**	**500**	**600079**	**23396**	2842.88	76.694	1321.51	1500	23399	38
14	1708.06	**1708.06**	**0**	**1192.28**	**500**	**548953**	**30361**	2241.56	31.235	1616.09	600	26289	22
15	1858.89	**1858.89**	**0**	**1241.54**	**600**	**97328**	**14320**	8688.29	367.392	8067.04	600	4297	1031
16	1921.79	**1921.79**	**0**	**1513.11**	**400**	**785106**	**36241**	2254.57	17.316	1947.5	300	23496	19
17	1935.98	**1935.98**	**0**	**1428.02**	**500**	**2161292**	**70203**	2088.68	7.887	1474	600	50942	29
18	1947.84	**1947.84**	**0**	**1432.63**	**500**	**2075635**	**129395**	4297.36	120.622	3679.05	600	65972	44
19	2043.34	**2043.34**	**0**	**1534.27**	**500**	**684815**	**33171**	8757.32	328.578	8131	600	27287	19
20	2202.92	**2202.92**	**0**	**1795.59**	**400**	**3590285**	**143668**	2990.14	35.735	2359.84	600	59412	110
21	2210.43	**2210.43**	**0**	**1600.52**	**600**	**135547**	**17290**	2611.36	18.138	1992.33	600	4390	53
22	2238.84	**2238.84**	**0**	**1723.78**	**500**	**148803**	**14764**	2813.31	25.66	2197.28	600	3274	131
23	2257.88	**2257.88**	**0**	**1645.82**	**600**	**582332**	**24418**	2682.97	18.827	2055	600	27245	25
24	2280.01	**2280.01**	**0**	**1866.89**	**400**	**2558593**	**72960**	5180.62	127.22	4568.29	600	87168	18
25	2283.06	**2283.06**	**0**	**1669.04**	**600**	**7839**	**4995**	4783.59	109.526	4171.33	600	318	47
26	2293.67	**2293.67**	**0**	**1687.57**	**600**	**2014911**	**138780**	4231.44	84.484	3612.15	600	62842	68
27	2373.3	**2373.3**	**0**	**2064.81**	**300**	**743371**	**38296**	**2373.3**	**0**	**2064.81**	**300**	**23177**	**23**
28	2384.85	**2384.85**	**0**	**1872.06**	**500**	**3563194**	**113820**	3139.94	31.662	2511.6	600	105648	38
29	2386.1	**2386.1**	**0**	**1777.56**	**600**	**184748**	**16937**	3360.92	40.854	2740.85	600	8955	23
30	2455.93	**2455.93**	**0**	**1946.15**	**500**	**2127950**	**267007**	3927.2	59.906	2410.37	1500	7674	46
31	2477.92	**2477.92**	**0**	**1857.39**	**600**	**115802**	**16052**	**2477.92**	**0**	**1857.39**	**600**	**36719**	**5224**
32	2514.29	**2514.29**	**0**	**1899.12**	**600**	**669340**	**26893**	4225.23	68.049	3598.39	600	6753	150
33	2563.86	**2563.86**	**0**	**2061.5**	**500**	**2888812**	**246442**	3683.1	43.654	3051.82	600	21357	34
34	2566.68	**2566.68**	**0**	**1956.45**	**600**	**75026**	**12092**	3383.78	31.835	2764.29	600	3507	19
35	2592.86	**2592.86**	**0**	**2081.14**	**500**	**4851724**	**131544**	3487.19	34.492	3071.47	400	184429	1442
36	2631.11	**2631.11**	**0**	**2122.3**	**500**	**692365**	**33123**	3310.18	25.809	2683.86	600	27404	24
37	2664.18	**2664.18**	**0**	**2150.42**	**500**	**7166845**	**230669**	3457.29	29.769	2825.5	600	187427	19
38	2715.06	**2715.06**	**0**	**2311.91**	**400**	**2980839**	**161196**	3441.8	26.767	3135.12	300	25052	21
39	2795	**2795**	**0**	**2285.53**	**500**	**10083492**	**329612**	5353.59	91.542	2941.5	2400	22895	36
40	2811.14	**2811.14**	**0**	**2197.64**	**600**	**12447466**	**354094**	6052.6	115.307	5433.62	600	133085	39
41	2835.55	**2835.55**	**0**	**2228.88**	**600**	**130433**	**15000**	3110.15	9.684	2502.05	600	9001	125
42	2854.99	**2854.99**	**0**	**2444.41**	**400**	**883430**	**36431**	5032.16	76.258	4405.18	600	29453	22
43	2857	**2857**	**0**	**2440.62**	**400**	**3246210**	**183333**	3929.82	37.55	3298.54	600	59749	117
44	2887.82	**2887.82**	**0**	**2278.61**	**600**	**993371**	**46333**	4277.47	48.121	3653.42	600	32171	18
45	2913.16	**2913.16**	**0**	**2305.51**	**600**	**125717**	**16694**	20161.93	592.098	19540.23	600	2880	26
46	2923.8	**2923.8**	**0**	**2311.98**	**600**	**901799**	**40077**	5427.32	85.626	4799.35	600	29495	18
47	2930.63	**2930.63**	**0**	**2314.93**	**600**	**9023**	**4464**	3246.9	10.792	2632.23	600	482	56
48	3014.12	**3014.12**	**0**	**2398.92**	**600**	**106140**	**14233**	3061.08	1.558	2441.84	600	11243	50
49	3031.23	**3031.23**	**0**	**2522.27**	**500**	**778854**	**36693**	5735.8	89.224	4215.19	1500	30596	22
50	3069.39	**3069.39**	**0**	**2553.81**	**500**	**66155**	**11373**	3222.44	4.986	2706.86	500	4499	973
51	3107.31	**3107.31**	**0**	**2496.68**	**600**	**811783**	**41640**	4683.3	50.719	4057.83	600	27841	17
52	3138.39	**3138.39**	**0**	**2513.57**	**600**	**671503**	**29864**	3906.52	24.475	3292.31	600	24684	103
53	3147.74	**3147.74**	**0**	**2539.71**	**600**	**131990**	**17466**	3705.33	17.714	3083.82	600	7820	22
54	3216.34	**3216.34**	**0**	**2708.97**	**500**	**229329**	**17845**	3633.94	12.984	3012.44	600	7995	19
55	3334.74	**3334.74**	**0**	**2724.24**	**600**	**7848**	**4750**	6763.54	102.82	6148.87	600	294	23
56	3350.28	**3350.28**	**0**	**2841.16**	**500**	**677749**	**28588**	4110.63	22.695	3485.81	600	23253	39
57	3377.63	**3377.63**	**0**	**2761.97**	**600**	**98414**	**16426**	4330.57	28.213	3711.54	600	3826	15
58	3387.26	**3387.26**	**0**	**2771.24**	**600**	**1254045**	**91902**	10297.95	204.02	9681.61	600	17330	1641
59	3392.41	**3392.41**	**0**	**2783.47**	**600**	**128887**	**16427**	4290.75	26.481	3672.7	600	4749	38
60	3469.1	**3469.1**	**0**	**2861.07**	**600**	**3096415**	**214517**	4324.66	24.662	4010.67	300	71134	21
61	3516.59	**3516.59**	**0**	**3013.88**	**500**	**8319328**	**281172**	4306.48	22.462	3685.36	600	187376	88
62	3566.43	**3566.43**	**0**	**3056.63**	**500**	**151912**	**18104**	4365.12	22.395	3744.96	600	10827	37
63	3590.25	**3590.25**	**0**	**2977.45**	**600**	**773547**	**34017**	4530.67	26.194	3903.37	600	26300	17
64	3602.87	**3602.87**	**0**	**3091.96**	**500**	**1146525**	**110899**	4140.12	14.912	3522.4	600	15510	138
65	3667.43	**3667.43**	**0**	**3053.58**	**600**	**562455**	**27071**	6164.37	68.084	5537.07	600	31928	23
66	3715.52	**3715.52**	**0**	**3102.74**	**600**	**5007359**	**178855**	5539.01	49.078	4913.73	600	27488	139
67	3732.89	**3732.89**	**0**	**3224.98**	**500**	**724207**	**37603**	4942.94	32.416	4318.58	600	26449	17
68	3745.23	**3745.23**	**0**	**3242.93**	**500**	**3788842**	**106805**	4747.63	26.765	4438.6	300	119203	19
69	3756.51	**3756.51**	**0**	**3145.11**	**600**	**779656**	**37075**	4648.85	23.755	4021.55	600	25774	21
70	3774.6	**3774.6**	**0**	**3265.33**	**500**	**11247**	**5093**	3890.15	3.061	3275.48	600	479	20
71	3792.84	**3792.84**	**0**	**3182.27**	**600**	**2831754**	**125522**	**3792.84**	**0**	**3182.27**	**600**	**60724**	**836**
72	3805.93	**3805.93**	**0**	**3397.75**	**400**	**228330**	**17771**	4071.49	6.978	3450.44	600	9410	28
73	3849.32	**3849.32**	**0**	**3231.33**	**600**	**133489**	**17392**	4789.42	24.422	4169.73	600	8851	24
74	3870.37	**3870.37**	**0**	**3256.76**	**600**	**124333**	**15909**	24894.22	543.199	24387.49	500	2906	61
75	4029.74	**4029.74**	**0**	**3421.22**	**600**	**679944**	**30296**	7887.82	95.74	7369.96	500	44850	2308
76	4038.52	**4038.52**	**0**	**3428.94**	**600**	**126029**	**16375**	15687.79	288.454	15066.09	600	4431	32
77	4049.36	**4049.36**	**0**	**3434.36**	**600**	**6305**	**4315**	25832.03	537.929	25219.38	600	741	689
78	4095.89	**4095.89**	**0**	**3484.21**	**600**	**1190491**	**95442**	5516.29	34.679	4891.99	600	6706	30
79	4129.25	**4129.25**	**0**	**3514.12**	**600**	**823788**	**32530**	5483.93	32.807	4858.14	600	32488	43
80	4131.07	**4131.07**	**0**	**3623.97**	**500**	**874254**	**42736**	5265.81	27.468	4642.3	600	33766	249
81	4345.84	**4345.84**	**0**	**3738.71**	**600**	**114030**	**15082**	8020.4	84.553	7398.89	600	6897	22
82	4353.37	**4353.37**	**0**	**3846.83**	**500**	**188243**	**14486**	4941.08	13.5	4320.42	600	3200	22
83	4370.79	**4370.79**	**0**	**3760.54**	**600**	**8121**	**4359**	15172.85	247.142	14561.63	600	488	45
84	4512.67	**4512.67**	**0**	**4000.07**	**500**	**3396180**	**137266**	6676.21	47.944	5154.05	1500	65647	57
85	4560.18	**4560.18**	**0**	**3947.66**	**600**	**91962**	**13554**	24808.2	444.018	24197.11	600	4165	836
86	4702.33	**4702.33**	**0**	**4094.24**	**600**	**622500**	**31067**	5997.58	27.545	4489.61	1500	27324	52
87	4873.87	**4873.87**	**0**	**4266.74**	**600**	**113225**	**15689**	6333.05	29.939	5711.04	600	4112	23
88	4924.57	**4924.57**	**0**	**4419.61**	**500**	**177896**	**16376**	6304.95	28.031	5682.48	600	8698	16
89	4966.52	**4966.52**	**0**	**4661.47**	**300**	**806739**	**36427**	**4966.52**	**0**	**4661.47**	**300**	**23971**	**20**
90	4970.95	**4970.95**	**0**	**4360.57**	**600**	**160311**	**16357**	6771.91	36.23	6152.23	600	8133	20
91	5010.01	**5010.01**	**0**	**4399.9**	**600**	**112070**	**14787**	34541.84	589.456	33925.47	600	6862	1226
92	5228.19	**5228.19**	**0**	**4617.71**	**600**	**1012601**	**61694**	6828.18	30.603	6202.39	600	22651	27
93	5335.97	**5335.97**	**0**	**4728.58**	**600**	**3186101**	**228707**	7207.77	35.079	6577.15	600	55041	22
94	5459.85	**5459.85**	**0**	**4849.62**	**600**	**107125**	**13948**	6471.12	18.522	5851.63	600	4211	50
95	5924.78	**5924.78**	**0**	**5308.02**	**600**	**92189**	**14126**	21188.94	257.633	20566.92	600	16929	2812
96	5926.39	**5926.39**	**0**	**5315.97**	**600**	**124280**	**14944**	7226.59	21.939	6605.54	600	8118	17
97	5974.83	**5974.83**	**0**	**5468.81**	**500**	**9885913**	**565939**	7334.68	22.759	5920.57	1400	22133	43
98	6250.1	**6250.1**	**0**	**5636.94**	**600**	**976954**	**39952**	7644.34	22.307	7018.02	600	28667	23
99	6599.83	**6599.83**	**0**	**5986.97**	**600**	**2437463**	**98951**	8822.9	33.684	8193.64	600	69032	305
100	6798.9	**6798.9**	**0**	**6182.15**	**600**	**98447**	**14071**	22839.28	235.926	22217.26	600	4661	778

Table 5.4 – Results for the FWRP with MILP on $B^{30^\circ}_{cat.}$ instances.

Chapter 6

Resolution of the FWRP using Informed Search Algorithms

In this chapter, a formulation adapted to ISAs is proposed for the FWRP. This formulation is based on the notion of routing plan, introduced in Section 6.1, that describes a partially routed waveguide. The feasibility of a routing plan can be evaluated using Linear Programming (LP) and its successors easily expressed through routing decisions like the addition of a bend. From this formulation, the ISAs presented in the state of the art in Section 3 can be used to solve the FWRP. To do so, two different heuristic evaluation functions are proposed in Section 6.2. Last, the Search Problem (SP) formulation is experimented in Section 6.3 on the previously introduced test instances.

6.1 Routing plan formulation

In order to route a waveguide π starting from its origin configuration $\theta^{ori.}$, its neutral fibre $\mathcal{F}_\pi$ can be built iteratively by making at each step decisions like adding a new bend from catalogue $B_{cat.}$ at the end of the waveguide. To formalise this approach, the concept of routing plan is introduced to represent the decisions made so far on the waveguide components.

6.1.1 Routing plan

Definition 20: Routing plan (in free space)

In free space, a *routing plan* s describes, in an abstract way, a neutral fibre $\mathcal{F}_s$ composed of N_s successive segments with, for each segment $k \in [\![1, N_s - 1]\!]$, the orientation change $r_k^s \in \mathcal{R}_k$ applied at the end point of segment k (see Figure 6.1). Moreover, a routing plan can be terminated or not, characterised by a boolean $Term^s$. When routing plan s is terminated, neutral fibre $\mathcal{F}_s$ has to reach the destination polyhedron $\mathcal{P}^{dest.}$.
The set of routing plans is denoted by $\mathcal{S}$.

Several data can be derived from the basic definition of a routing plan $s \in \mathcal{S}$. Especially, bends and orientations of the partial waveguide can be deduced from the sequence of orientation changes. The bend $b_k^s \in B_{cat.}$ applied at the end point of the k^{th} segment of neutral fibre $\mathcal{F}_s$ is the bend $b_{r_k^s}$ for $k \in [\![1, N_s - 1]\!]$. In the same way, the orientation $o_k^s \in \mathcal{O}_\infty$ of the k^{th} segment is the orientation $o^+_{r^s_{k-1}}$ for $k \in [\![2, N_s]\!]$. The first orientation is $o_1^s = o^{ori.}$.

Routing plan s		
N_s	:	Size
$\left[r^s_1, ..., r^s_{N_s-1}\right]$	:	Orientation changes
$Term^s$	:	$true/false$
$\left[b^s_1, ..., b^s_{N_s-1}\right]$	:	Bends
$\left[o^s_1, ..., o^s_{N_s}\right]$	:	Orientations

(a) Description of plan s.

(b) Solution of LP^L_s.

Figure 6.1 – A routing plan $s \in \mathcal{S}$.

6.1.2 Feasibility and cost-from-origin

A routing plan $s \in \mathcal{S}$ is *feasible* if it is possible to create a neutral fibre following the choices made in s and satisfying the waveguide routing constraints introduced in Section 4.2.1 on page 45. This feasibility problem can be formulated as a linear program, referred to as LP^L_s, that contains two kinds of variables:

- *position variables* $p^s_k = \left(p^s_{k,x}, p^s_{k,y}, p^s_{k,z}\right)$ such that, for $k \in [\![1, N_s + 1]\!]$, real variable $p^s_{k,x}$ (respectively $p^s_{k,y}$ and $p^s_{k,z}$) is the x-coordinate (respectively y-coordinate and z-coordinate) of the k^{th} point of the neutral fibre;
- *length variables* ℓ^s_k such that, for $k \in [\![1, N_s]\!]$, real variable ℓ^s_k is the length of the k^{th} segment of the neutral fibre, or in other words $\ell^s_k = \left\|\overrightarrow{p^s_k p^s_{k+1}}\right\|$.

Linear program LP^L_s minimises the cost of the partial neutral fibre which is described by the constraints of plan s. This cost is called *cost-from-origin* and is defined by $\sum_{k=1}^{N_s-1} \gamma_{b^s_k} + \mu \sum_{k=1}^{N_s} \ell^s_k$. Thus, linear program LP^L_s can be formulated as follows:

$$\text{minimise } \sum_{k=1}^{N_s-1} \gamma_{b^s_k} + \mu \sum_{k=1}^{N_s} \ell^s_k \tag{6.1}$$

subject to:

$$p^s_1 \in \mathcal{P}^{ori.} \tag{6.2}$$

$$p^s_{N_s+1} \in \mathcal{P}^{dest.} \qquad \text{if } Term^s = true \tag{6.3}$$

$$\ell^s_k \geq L_{b^s_{k-1}} + L^{min} + L_{b^s_k} \qquad \forall k \in [\![1, N_s]\!] \tag{6.4}$$

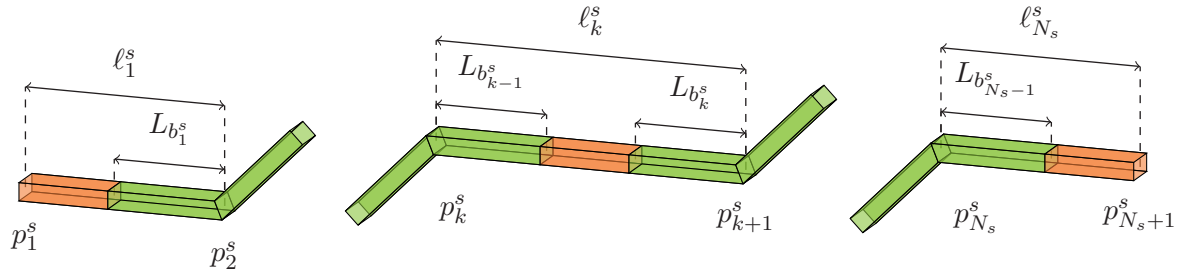
$$\overrightarrow{p^s_k p^s_{k+1}} = \ell^s_k \overrightarrow{e_{o^s_k, z}} \qquad \forall k \in [\![1, N_s]\!] \tag{6.5}$$

$$\ell^s_k \in \mathbb{R}^+ \qquad \forall k \in [\![1, N_s]\!] \tag{6.6}$$

$$p^s_k \in \mathbb{R}^3 \qquad \forall k \in [\![1, N_s + 1]\!] \tag{6.7}$$

Constraint 6.2 states that the neutral fibre must start from the origin polyhedron. Such an inclusion constraint within a convex polyhedron can be expressed as a set of linear constraints. Constraint 6.3 imposes that the neutral fibre must reach the destination polyhedron if plan s is terminated. Constraints 6.4 impose a minimal length on the segments. For the k^{th} segment, this minimum length is obtained from the minimal length L^{min} of straight sections and from the respective contributions $L_{b^s_{k-1}}$ and $L_{b^s_k}$ of the previous and next bends (see Figure 6.2). By convention, it is assumed that $L_{b^s_0} = 0$ and $L_{b^s_{N_s}} = 0$, since the first and last segments do not have a previous or a next bend respectively. Constraints 6.5 define the coordinates of the $(k+1)^{\text{th}}$ point of the neutral fibre from the coordinates of the k^{th} point, the orientation of the k^{th} segment as specified by routing plan s, and the length of this segment.

If LP^L_s has a solution, then routing plan s is *feasible*. In this case, the optimal cost-from-origin of linear program LP^L_s is a lower bound of the cost of any waveguide that satisfies the



(a) First segment. (b) Intermediate segment. (c) Last segment.

Figure 6.2 – Portion of a waveguide between two successive break points of the neutral fibre $\mathcal{F}_s$ (the straight section with variable length is depicted in orange).

constraints defined by routing plan s. This value, referred to as $g_L(s)$, is called the *lazy cost-from-origin* because it corresponds to a neutral fibre with a minimal length which somehow stays as close as possible to the origin. For this reason, LP_s^L is called *lazy linear program* in the remainder of this thesis. When $Term^s = true$, the lazy cost-from-origin corresponds to the optimal cost of a waveguide which uses the bends of routing plan s. The solution waveguide can be rebuilt with Algorithm 10 on page 59 using the optimal origin position and segment lengths found by LP_s^L, as well as the bends already defined by plan s. On the contrary, when there is no solution, then routing plan s is not feasible and, by convention, its minimal cost-from-origin is infinite, that is to say $g_L(s) = \infty$.

6.1.3 Neighbourhood

A non-terminated routing plan $s \in \mathcal{S}$ in free space can be extended using two kinds of decisions detailed below: add a bend or finish the plan. So the successors of a routing plan are built by forward chaining.

It is important to note that with these elementary decisions, there is a single way to reach any routing plan $s \in \mathcal{S}$, which is to apply exactly the same sequence of bends and terminate the plan if needed. So, during an exploration of routing plan space $\mathcal{S}$, it is impossible to visit a given plan s more than once by starting from an initial plan and taking successive decisions.

Add a bend

Routing plan s'
$N_{s'} = N_s + 1$
$\left[r_1^s, ..., r_{N_s-1}^s, r\right]$
$false$
$\left[b_1^s, ..., b_{N_s-1}^s, b_r\right]$
$\left[o_1^s, ..., o_{N_s}^s, o_r^+\right]$

(a) Description of successor s'.

(b) Solution of $LP_{s'}^L$.

Figure 6.3 – Adding an orientation change $r \in \mathcal{R}_\infty^{out.}\left(o_{N_s}^s\right)$ to a plan $s \in \mathcal{S}$.

If the maximum number of bends is not reached, that means $N_s < N_S$, and if $r \in \mathcal{R}_\infty^{out.}\left(o_{N_s}^s\right)$ is a reachable orientation change such that $N_s + minBends\left(o_r^+\right) \leq N_S$ (the maximum number of segments is not exceeded), then it is possible to apply bend b_r at the end of the last segment of routing plan s. Note that such orientation changes are described by the k-candidate orientation

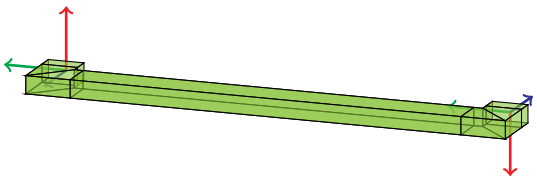
change set $\mathcal{R}^{out.}_{N_s}\left(o^s_{N_s}\right)$ but, in practice, the enumeration of candidate orientation changes is avoided when using the ISA formulation. Let $s' \in \mathcal{S}$ be the successor routing plan resulting from the addition of a bend. Formally, adding bend b_r associated with orientation change r creates a new segment in the neutral fibre, that is to say $N_{s'} = N_s + 1$, and extends the sequence of orientation changes with r, meaning that $r^{s'}_k = r^s_k$ for $k \in [\![1, N_s - 1]\!]$ and $r^{s'}_{N_s} = r$. The successor s' is described on Figure 6.3a and illustrated on Figure 6.3b on the previous page.

Finish a plan

If a destination orientation has been reached, that is to say $o^s_{N_s} \in \mathcal{O}^{dest.}$, then routing plan s can be terminated as illustrated on Figure 10.5a. Its successor s' is the same plan but terminated, meaning that $Term^{s'} = true$, and it has to reach the destination polyhedron $\mathcal{P}^{dest.}$ (see Constraint 6.3). Remind that the resulting terminated plan may be infeasible and, in this case, it is not considered as a valid successor of plan s.

Routing plan s'
$N_{s'} = N_s$
$\left[r^s_1, ..., r^s_{N_s-1}\right]$
$true$
$\left[b^s_1, ..., b^s_{N_s-1}\right]$
$\left[o^s_1, ..., o^s_{N_s}\right]$

(a) Description of successor s'.



(b) Solution of $LP^L_{s'}$.

Figure 6.4 – Terminating a plan $s \in \mathcal{S}$.

6.1.4 Origin and destination plans

Thanks to the routing plans presented above, the FWRP can be reformulated as finding a routing plan $s \in \mathcal{S}$ that connects the origin and destination polyhedrons, satisfies the constraints introduced in Section 4.2.1 and minimises the waveguide cost presented in Section 4.2.2. Such a routing plan is called a *destination plan*. It is a routing plan $s \in \mathcal{S}$ that is feasible, terminated and for which the last orientation is a destination orientation, that means $o^s_{N_s} \in \mathcal{O}^{dest.}$. The set of destination plans is referred to as $\mathcal{S}_{dest.}$. To explore the space of routing plans $\mathcal{S}$, routing decisions like the addition of a bend are applied from an *origin plan* referred to as $s_{ori.}$. For a FWRP instance, plan $s_{ori.}$ starts from the origin orientation $o^{ori.}$ and does not contain any bend, that means $N_{s_{ori.}} = 1$, as shown on Figure 6.5a. Of course, it is a non-terminated plan.

Routing plan $s_{ori.}$
$N_s = 1$
$\varnothing$
$false$
$\varnothing$
$\left[o^{ori.}\right]$

(a) Origin plan $s_{ori.}$.

Routing plan s
$N_s \leq N_S$
$\left[r^s_1, ..., r^s_{N_s-1}\right]$
$true$
$\left[b^s_1, ..., b^s_{N_s-1}\right]$
$\left[o^s_1, ..., o^s_{N_s}\right]$ with $o^s_{N_s} \in \mathcal{O}^{dest.}$

(b) Destination plan $s \in \mathcal{S}_{dest.}$.

Figure 6.5 – Origin and destination plans.

6.2 Evaluations and heuristics

Section 6.1 described the space of routing plans $\mathcal{S}$, the origin and destination plans $s_{ori.}$ and $\mathcal{S}_{dest.}$, as well as the neighbourhood of a routing plan $s \in \mathcal{S}$. However, in order to solve the FWRP with ISAs, it is necessary to define a heuristic function $h(s)$, which estimates the remaining cost to reach a destination plan in $\mathcal{S}_{dest.}$ from feasible plan $s \in \mathcal{S}$, or an evaluation function $f(s)$, which estimates the cost of a destination plan in $\mathcal{S}_{dest.}$ extended from feasible plan $s \in \mathcal{S}$.

6.2.1 Partial cost-to-destination

As explained in Section 6.1.2, a *partial cost-from-origin* can be defined using a feasibility problem for any feasible routing plan $s \in \mathcal{S}$. It corresponds to the cost of a partial neutral fibre that must satisfy the constraints of plan s. Similarly, it is possible to estimate a *partial cost-to-destination* of the partial neutral fibre that is not already constrained by plan s. To do so, a relaxed polyline is built to connect the last point of the current partial neutral fibre to the destination polyhedron $\mathcal{P}^{dest.}$ by ignoring the orientation constraints on the segments of the polyline.

Two cost-to-destination evaluation methods are proposed depending on the partial neutral fibre considered. The first approach, presented in Section 6.2.2, reuses the polyline provided by lazy linear program LP_s^L which minimises the partial cost-from-origin while the second method, introduced in Section 6.2.3, is based on a reformulation of the feasibility problem into a destination-attracted linear program LP_s^D which minimises the partial cost-to-destination.

In both cases, the partial cost-to-destination can be divided into two parts, similarly to the cost of a waveguide: the bend cost and the linear cost.

Partial bend cost-to-destination

On the one hand, the partial cost-to-destination must take into account the cost of the bend combination that must be added to routing plan s in order to reach a destination orientation in $\mathcal{O}^{dest.}$ and the destination polyhedron $\mathcal{P}^{dest.}$. A lower bound on the remaining bend cost is the minimal cost of a bend combination that allows reaching a destination orientation in $\mathcal{O}^{dest.}$ from the current orientation $o^s_{N_s}$ in the kernel of reachable orientations $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$. This minimal bend combination cost, referred to as $\gamma_{min.}(o)$, can be precomputed for any reachable orientation $o \in \mathcal{O}_\infty$ by applying Algorithm 8 with the cost function $\gamma(r) = \gamma_{b_r}$ defined for any orientation change $r \in \mathcal{R}_\infty$. Figure 6.6 illustrates this preprocessing for the example of Figure 5.1 with $B_{cat.} = \{b_1, b_2\}$, $o^{ori.} = o_1$ and $\mathcal{O}^{dest.} = \{o_4\}$.

Partial linear cost-to-destination

On the other hand, the partial cost-to-destination must also consider the linear cost coming from the length of the neutral fibre that connects the last point $p^s_{N_s+1}$ of a feasible routing plan $s \in \mathcal{S}$ to the destination polyhedron $\mathcal{P}^{dest.}$. Since the simplest way to connect two points is to use a straight line, the partial linear cost-to-destination can be evaluated as the minimal value of the *as the crow flies distance* between the last point $p^s_{N_s+1}$ and the destination polyhedron $\mathcal{P}^{dest.}$ weighted by linear cost μ.

If destination polyhedron $\mathcal{P}^{dest.}$ is a point P^{dest}, it simply corresponds to $\mu \left\| \overrightarrow{p^s_{N_s+1} P^{dest}} \right\|$. Otherwise, finding the minimal distance $\min_{p \in \mathcal{P}^{dest.}} \left\| \overrightarrow{p^s_{N_s+1} p} \right\|$ is a quadratic programming problem which is time-consuming if solved numerous times. In practice, rather than solving this quadratic program, several points are sampled in destination polyhedron $\mathcal{P}^{dest.}$ and only the minimal distance to the sampled points is considered.

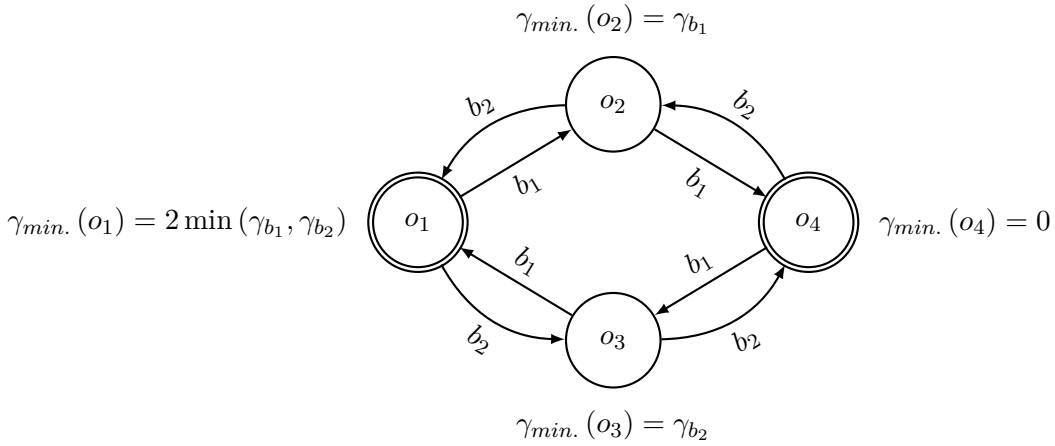


Figure 6.6 – Minimal costs to the destination orientations.

6.2.2 Lazy approach

A first way to build the partial neutral fibre $\mathcal{F}_s$ for a feasible routing plan $s \in \mathcal{S}$ is to use the polyline $\left[p_1^s, ..., p_{N_s+1}^s\right]$ provided by lazy linear program LP_s^L which minimises the cost-from-origin. As illustrated on Figure 6.7, the partial neutral fibre stays as close as possible to the origin polyhedron $\mathcal{P}^{ori.}$ in order to minimise the cost-from-origin. Therefore, the cost-from-origin corresponds to the lazy cost $g_L(s)$ and the cost-to-destination using the polyline $\left[p_1^s, ..., p_{N_s+1}^s\right]$ provided by LP_s^L can be evaluated with the *lazy heuristic* $h_L(s)$ defined as follows.

Definition 21: Lazy heuristic h_L

Let s be a feasible routing plan in $\mathcal{S}$ and $\left[p_1^s, ..., p_{N_s+1}^s\right]$ the polyline provided by lazy linear program LP_s^L which minimises the cost-from-origin. The *lazy heuristic* is defined for routing plan s by:

$$h_L(s) = \begin{cases} \gamma_{min.}\left(o_{N_s}^s\right) + \mu \min_{p \in \mathcal{P}^{dest.}} \left\| \overrightarrow{p_{N_s+1}^s p} \right\| & \text{if } Term^s = false \\ 0 & \text{otherwise} \end{cases}$$

Figure 6.7 – Partial neutral fibre and polyline to the destination with the lazy approach.

Optimal solutions of the lazy linear program

When routing plan $s \in \mathcal{S}$ is not terminated (when $Term^s = false$), it is possible to directly compute the optimal solution of linear program LP_s^L without applying the simplex algorithm by setting each segment of the polyline to its minimal length.

Property 3: Optimal solution of non-terminated routing plans

Let s be a non-terminated routing plan in $\mathcal{S}$ (such that $Term^s = false$) and $P^{ori.}$ a point in the origin polyhedron $\mathcal{P}^{ori.}$. The segment lengths ℓ_k^s, for $k \in [\![1, N_s]\!]$, and positions p_k^s, for $k \in [\![1, N_s + 1]\!]$, defined as follows form an optimal solution of lazy linear program LP_s^L (remind that $L_{b_0^s} = 0$ by convention):

$$\begin{aligned} \ell_k^s &= L_{b_{k-1}^s} + L^{min} + L_{b_k^s} \quad \forall k \in [\![1, N_s]\!] \\ p_1^s &= P^{ori.} \\ \overrightarrow{p_k^s p_{k+1}^s} &= \ell_k^s \overrightarrow{e_{o_k^s, z}} \quad \forall k \in [\![1, N_s]\!] \end{aligned}$$

Proof: *By definition, segment lengths ℓ_k^s and positions p_k^s defined by Property 3 form a solution that satisfies all the constraints of lazy linear program LP_s^L. Assume that segment lengths $\ell_k^{s'}$ and positions $p_k^{s'}$ form another solution of LP_s^L. Then, for $k \in [\![1, N_s]\!]$, segment length $\ell_k^{s'}$ verifies:*

$$\begin{aligned} \ell_k^{s'} &\geq L_{b_{k-1}^s} + L^{min} + L_{b_k^s} \\ &= \ell_k^s \end{aligned}$$

It results that:

$$\sum_{k=1}^{N_s - 1} \gamma_{b_k^s} + \mu \sum_{k=1}^{N_s} \ell_k^{s'} \geq \sum_{k=1}^{N_s - 1} \gamma_{b_k^s} + \mu \sum_{k=1}^{N_s} \ell_k^s$$

Q.E.D.

As a consequence, a non-terminated plan $s \in \mathcal{S}$ is always feasible and its lazy cost $g_L(s)$ can be easily computed.

Property 4: Lazy cost of non-terminated routing plans

Let s be a non-terminated routing plan in $\mathcal{S}$. The lazy cost $g_L(s)$ of routing plan s is defined by:

$$g_L(s) = \mu N_s L^{min} + \sum_{k=1}^{N_s - 1} \left(\gamma_{b_k^s} + 2\mu L_{b_k^s} \right)$$

Monotonicity of the lazy cost

Property 5: Monotonicity of the segment lengths

Let s be a non-terminated routing plan in $\mathcal{S}$ and s' a descendant of plan s. For $k \in [\![1, N_s]\!]$, the optimal segment length $\ell_k^{s'}$ provided by $LP_{s'}^L$ satisfies:

$$\ell_k^{s'} \geq \ell_k^s$$

where ℓ_k^s is provided by LP_s^L.

Proof: *Since routing plan s is non-terminated, the optimal segment length ℓ_k^s provided by linear program LP_s^L is $L_{b_{k-1}^s} + L^{min} + L_{b_k^s}$. By definition of linear program $LP_{s'}^L$, the optimal segment length $\ell_k^{s'}$ is lower-bounded by $L_{b_{k-1}^s} + L^{min} + L_{b_k^s}$. Q.E.D.*

Property 6: Monotonicity of the lazy cost g_L

Let s be a non-terminated routing plan in $\mathcal{S}$ and s' a descendant of plan s. The lazy cost function g_L is monotonic and verifies:

$$g_L(s) \leq g_L(s')$$

Proof: *Since plan s' is a descendant of plan s, then $N_{s'} \geq N_s + 1$. Using Property 5 and the positivity of segment lengths, it is shown that:*

$$\begin{aligned} \sum_{k=1}^{N_s} \ell_k^s &\leq \sum_{k=1}^{N_s} \ell_k^{s'} \\ &\leq \sum_{k=1}^{N_{s'}} \ell_k^{s'} \end{aligned}$$

Moreover, since plan s' is a descendant of plan s, the bends of plan s' satisfy $b_k^{s'} = b_k^s$, for $k \in [\![1, N_s - 1]\!]$. Using the positivity of bend costs, it is shown that:

$$\begin{aligned} \sum_{k=1}^{N_s} \gamma_{b_k^s} &= \sum_{k=1}^{N_s} \gamma_{b_k^{s'}} \\ &\leq \sum_{k=1}^{N_{s'}} \gamma_{b_k^{s'}} \end{aligned}$$

Finally, it results that:

$$\mu \sum_{k=1}^{N_s} \ell_k^s + \sum_{k=1}^{N_s} \gamma_{b_k^s} \leq \mu \sum_{k=1}^{N_{s'}} \ell_k^{s'} + \sum_{k=1}^{N_{s'}} \gamma_{b_k^{s'}}$$

In other words, $g_L(s) \leq g_L(s')$.

Consistency of the lazy heuristic

Property 7: Upper bound on the distance to a descendant's last point

Let s be a non-terminated routing plan in $\mathcal{S}$ and s' a descendant of plan s. The distance between the last point $p^s_{N_s+1}$ of plan s and the last point $p^{s'}_{N_{s'}+1}$ of plan s' is upper bounded by:

$$\left\|\overrightarrow{p^s_{N_s+1}p^{s'}_{N_{s'}+1}}\right\| \leq \sum_{k=1}^{N_s}\left(\ell^{s'}_k - \ell^s_k\right) + \left\|\overrightarrow{p^s_1 p^{s'}_1}\right\| + \sum_{k=N_s+1}^{N_{s'}} \ell^{s'}_k$$

where p^s_k and $p^{s'}_k$ as well as ℓ^s_k and $\ell^{s'}_k$ are respectively provided by lazy linear programs LP^L_s and $LP^L_{s'}$.

Proof: *Using Constraint 6.5, it can be shown by induction that* $\overrightarrow{p^s_1 p^s_{k'}} = \sum_{k=1}^{k'-1} \ell^s_k \overrightarrow{e_{o^s_k,z}}$ *for all* $s \in \mathcal{S}$ *and* $k' \in [\![1, N_s+1]\!]$. *Then, it results that:*

$$\begin{aligned}\left\|\overrightarrow{p^s_{N_s+1}p^{s'}_{N_{s'}+1}}\right\| &= \left\|\overrightarrow{p^s_{N_s+1}p^s_1} + \overrightarrow{p^s_1 p^{s'}_1} + \overrightarrow{p^{s'}_1 p^{s'}_{N_{s'}+1}}\right\| \\ &= \left\|-\sum_{k=1}^{N_s}\ell^s_k\overrightarrow{e_{o^s_k,z}} + \overrightarrow{p^s_1 p^{s'}_1} + \sum_{k=1}^{N_{s'}}\ell^{s'}_k\overrightarrow{e_{o^{s'}_k,z}}\right\|\end{aligned}$$

However, since routing plan s' *is a descendant of plan* s, *the orientations of plan* s' *satisfy* $o^{s'}_k = o^s_k$ *for* $k \in [\![1, N_s]\!]$. *Therefore:*

$$\left\|\overrightarrow{p^s_{N_s+1}p^{s'}_{N_{s'}+1}}\right\| = \left\|\sum_{k=1}^{N_s}\left(\ell^{s'}_k - \ell^s_k\right)\overrightarrow{e_{o^s_k,z}} + \overrightarrow{p^s_1 p^{s'}_1} + \sum_{k=N_s+1}^{N_{s'}}\ell^{s'}_k\overrightarrow{e_{o^{s'}_k,z}}\right\|$$

Using the sublinearity of the Euclidean norm, it results that:

$$\left\|\overrightarrow{p^s_{N_s+1}p^{s'}_{N_{s'}+1}}\right\| \leq \sum_{k=1}^{N_s}\left|\ell^{s'}_k - \ell^s_k\right| + \left\|\overrightarrow{p^s_1 p^{s'}_1}\right\| + \sum_{k=N_s+1}^{N_{s'}}\left|\ell^{s'}_k\right|$$

Last, using the positivity of the segment lengths and Property 5, it shows that:

$$\left\|\overrightarrow{p^s_{N_s+1}p^{s'}_{N_{s'}+1}}\right\| \leq \sum_{k=1}^{N_s}\left(\ell^{s'}_k - \ell^s_k\right) + \left\|\overrightarrow{p^s_1 p^{s'}_1}\right\| + \sum_{k=N_s+1}^{N_{s'}}\ell^{s'}_k$$

Property 8: Consistency of the lazy heuristic h_L

If the origin polyhedron $\mathcal{P}^{ori.}$ is a point, then the lazy heuristic h_L is consistent, that is to say $h_L\left(s^{dest.}\right) = 0$ for all destination routing plans $s^{dest.} \in \mathcal{S}_{dest.}$, and, for any non-terminated routing plan $s \in \mathcal{S}$ and any successor $s' \in \mathcal{S}$, it satisfies the triangular inequality:

$$h_L(s) \leq g_L(s') - g_L(s) + h_L(s')$$

Proof: *By definition, a destination routing plan* $s^{dest.} \in \mathcal{S}_{dest.}$ *is terminated and* $h_L\left(s^{dest.}\right) = 0$. *Now, let* s *be a non-terminated routing plan in* $\mathcal{S}$ *and* s' *a descendant of plan* s. *So, the bends of plan* s' *satisfy* $b_k^{s'} = b_k^s$, *for* $k \in [\![1, N_s - 1]\!]$. *Using Property 7, it results that:*

$$\begin{aligned} g_L(s') - g_L(s) + h_L(s') &= \sum_{k=1}^{N_{s'}-1} \gamma_{b_k^{s'}} + \mu \sum_{k=1}^{N_{s'}} \ell_k^{s'} - \sum_{k=1}^{N_s-1} \gamma_{b_k^s} - \mu \sum_{k=1}^{N_s} \ell_k^s \\ &\quad + \gamma_{min.}\left(o_{N_{s'}}^{s'}\right) + \mu \min_{p \in \mathcal{P}^{dest.}} \left\| \overrightarrow{p_{N_{s'}+1}^{s'} p} \right\| \\ &= \mu \sum_{k=1}^{N_s} \left(\ell_k^{s'} - \ell_k^s\right) + \mu \sum_{k=N_s+1}^{N_{s'}} \ell_k^{s'} \\ &\quad + \mu \min_{p \in \mathcal{P}^{dest.}} \left\| \overrightarrow{p_{N_{s'}+1}^{s'} p} \right\| + \sum_{k=N_s}^{N_{s'}-1} \gamma_{b_k^{s'}} + \gamma_{min.}\left(o_{N_{s'}}^{s'}\right) \\ &\geq \mu \left\| \overrightarrow{p_{N_s+1}^s p_{N_{s'}+1}^{s'}} \right\| - \mu \left\| \overrightarrow{p_1^s p_1^{s'}} \right\| \\ &\quad + \mu \min_{p \in \mathcal{P}^{dest.}} \left\| \overrightarrow{p_{N_{s'}+1}^{s'} p} \right\| + \sum_{k=N_s}^{N_{s'}-1} \gamma_{b_k^{s'}} + \gamma_{min.}\left(o_{N_{s'}}^{s'}\right) \end{aligned}$$

Applying the triangular inequality, it shows that:

$$\begin{aligned} g_L(s') - g_L(s) + h_L(s') &\geq \mu \min_{p \in \mathcal{P}^{dest.}} \left\| \overrightarrow{p_{N_s+1}^s p} \right\| - \mu \left\| \overrightarrow{p_1^s p_1^{s'}} \right\| \\ &\quad + \sum_{k=N_s}^{N_{s'}-1} \gamma_{b_k^{s'}} + \gamma_{min.}\left(o_{N_{s'}}^{s'}\right) \end{aligned}$$

Moreover, $\sum_{k=N_s}^{N_{s'}-1} \gamma_{b_k^{s'}} + \gamma_{min.}\left(o_{N_{s'}}^{s'}\right)$ *is the minimal bend cost of the best bend combination to reach a destination orientation in* $\mathcal{O}^{dest.}$ *which starts by bends* $b_{N_s}^{s'}$, ..., $b_{N_{s'}-1}^{s'}$. *By definition, this cost is lower bounded by the minimal bend cost* $\gamma_{min.}\left(o_{N_s}^s\right)$ *of the best bend combination to reach a destination orientation. Therefore:*

$$\begin{aligned} g_L(s') - g_L(s) + h_L(s') &\geq \mu \min_{p \in \mathcal{P}^{dest.}} \left\| \overrightarrow{p_{N_s+1}^s p} \right\| - \mu \left\| \overrightarrow{p_1^s p_1^{s'}} \right\| + \gamma_{min.}\left(o_{N_s}^s\right) \\ &= h_L(s) - \mu \left\| \overrightarrow{p_1^s p_1^{s'}} \right\| \end{aligned}$$

Finally, if the origin polyhedron $\mathcal{P}^{ori.}$ *is a point, then* $\left\| \overrightarrow{p_1^s p_1^{s'}} \right\| = 0$ *and* $g_L(s') - g_L(s) + h_L(s') \geq h_L(s)$.

Since the admissibility of a heuristic follows from its consistency, lazy heuristic h_L is admissible under the same conditions.

Property 9: Admissibility of the lazy heuristic h_L

If the origin polyhedron $\mathcal{P}^{ori.}$ is a point, then the lazy heuristic h_L is admissible in the sense that, for a routing plan $s \in \mathcal{S}$, $h_L(s)$ never overestimates the remaining cost to reach a destination plan in $\mathcal{S}_{dest.}$ from plan s.

The classical ISAs introduced in Section 3.2 are now applicable to solve the FWRP by using lazy cost function g_L, which is increasing (see Section 6.1.2), and lazy heuristic function h_L. For instance, Best-First Search approaches, which are complete algorithms, will provide results with interesting properties that may challenge the MILP formulation. Indeed, because of the consistency of lazy heuristic h_L, the first solution encountered by A* using $f_L(s) = g_L(s) + h_L(s)$ will be optimal, while the one found by WA* using $f_L(s) = g_L(s) + \epsilon h_L(s)$ will be guaranteed to cost no more than ϵ times the optimal cost. Furthermore, incomplete approaches like LSS-LRTA* with $f_L(s) = g_L(s) + h_L(s)$, and BrF-BS or even SAHC with $h_L(s)$ are also candidates to quickly solve the FWRP, even if they provide no guarantee on the solution quality. They will be tested and compared in Section 6.3.1.

6.2.3 Destination-attracted approach

Another approach consists in building a partial neutral fibre $\mathcal{F}_s$ for the feasible routing plan $s \in \mathcal{S}$ which gets as close as possible to the destination polyhedron $\mathcal{P}^{dest.}$. In this case, the evaluation difference between plans that quickly reach the destination and misleading ones will be accentuated. To do so, the feasibility problem is reformulated into a *destination-attracted linear program* LP_s^D presented in this section that minimises the Manhattan distance from the last point $p^s_{N_s+1}$ to the destination polyhedron $\mathcal{P}^{dest.}$.

As the Manhattan distance requires to compute the absolute value of the coordinate gap between $p^s_{N_s+1}$ and a goal point in $\mathcal{P}^{dest.}$ along each axis, the classical absolute value linearisation technique is used. Typically, if $x \in \mathbb{R}$ is a real variable whose absolute value $|x|$ must be minimised in the criterion, $|x|$ can be linearised by introducing two positive real variables $x^+ \in \mathbb{R}^+$ and $x^- \in \mathbb{R}^+$ that respectively correspond to the positive and negative part of x such that $|x|$ is replaced by $x^+ + x^-$. Then, x^+ and x^- are constrained with $x^+ \geq x$ and $x^- \geq -x$. Since the sum $x^+ + x^-$ is minimised, the constraints enforce one of the two variables x^+ and x^- to be equal to $|x|$ and the other to be zero.

Using this linearisation technique, destination-attracted linear program LP_s^D is written with the same variables and constraints as lazy linear program LP_s^L (see Section 6.1.2) but introduces the following additional variables:

- *goal position variables* $g^s = \left(g^s_x, g^s_y, g^s_z\right)$ such that real variable g^s_x (respectively g^s_y and g^s_z) is the x-coordinate (respectively y-coordinate and z-coordinate) of the goal point in destination polyhedron $\mathcal{P}^{dest.}$;

- *positive distance variables* d^{s+}_a, for $a \in \{x, y, z\}$, such that real variable d^{s+}_x (respectively d^{s+}_y and d^{s+}_z) is the positive part of $\left|p^s_{N_s+1,x} - g^s_x\right|$ (respectively $\left|p^s_{N_s+1,y} - g^s_y\right|$ and $\left|p^s_{N_s+1,z} - g^s_z\right|$);

- *negative distance variables* d^{s-}_a, for $a \in \{x, y, z\}$, such that real variable d^{s-}_x (respectively d^{s-}_y and d^{s-}_z) is the negative part of $\left|p^s_{N_s+1,x} - g^s_x\right|$ (respectively $\left|p^s_{N_s+1,y} - g^s_y\right|$ and $\left|p^s_{N_s+1,z} - g^s_z\right|$).

Linear program LP_s^D minimises the Manhattan distance from the last point $p_{N_s+1}^s$ to the destination polyhedron $\mathcal{P}^{dest.}$ that corresponds to value $d_x^{s+} + d_x^{s-} + d_y^{s+} + d_y^{s-} + d_z^{s+} + d_z^{s-}$. This approximatively minimises the *cost-to-destination* of the partial neutral fibre in the sense that the Manhattan distance is minimised rather than the Euclidean distance in order to satisfy linearity requirements. Finally, linear program LP_s^D can be formulated as follows:

$$\text{minimise } d_x^{s+} + d_x^{s-} + d_y^{s+} + d_y^{s-} + d_z^{s+} + d_z^{s-} \tag{6.8}$$

subject to:

$$p_1^s \in \mathcal{P}^{ori.} \tag{6.9}$$

$$p_{N_s+1}^s \in \mathcal{P}^{dest.} \qquad \text{if } Term^s = true \tag{6.10}$$

$$g^s \in \mathcal{P}^{dest.} \tag{6.11}$$

$$\ell_k^s \geq L_{b_{k-1}^s} + L^{min} + L_{b_k^s} \qquad \forall k \in [\![1, N_s]\!] \tag{6.12}$$

$$\overrightarrow{p_k^s p_{k+1}^s} = \ell_k^s \overrightarrow{e_{o_k^s, z}} \qquad \forall k \in [\![1, N_s]\!] \tag{6.13}$$

$$d_a^{s+} + g_a^s \geq p_{N_s+1,a}^s \qquad \forall a \in \{x, y, z\} \tag{6.14}$$

$$d_a^{s-} + p_{N_s+1,a}^s \geq g_a^s \qquad \forall a \in \{x, y, z\} \tag{6.15}$$

$$\ell_k^s \in \mathbb{R}^+ \qquad \forall k \in [\![1, N_s]\!] \tag{6.16}$$

$$p_{N_s}^s \in \mathbb{R}^3 \qquad \forall k \in [\![1, N_s + 1]\!] \tag{6.17}$$

$$d_a^{s+}, d_a^{s-} \in \mathbb{R}^+ \qquad \forall a \in \{x, y, z\} \tag{6.18}$$

Contrarily to lazy linear program LP_s^L which minimises the cost-from-origin, there is no obvious optimal solution to destination-attracted linear program LP_s^D and it must be solved to evaluate each routing plan $s \in \mathcal{S}$. The cost-from-origin corresponding to a solution of LP_s^D, called *destination-attracted cost* and referred to as $g_D(s)$ can be evaluated as follows.

Definition 22: Destination-attracted cost g_D

Let s be a feasible routing plan in $\mathcal{S}$ and $\ell_k^s, ..., \ell_{N_s}^s$ the segment lengths provided by destination-attracted linear program LP_s^D which minimises the partial cost-to-destination. The *destination-attracted cost* for routing plan s is defined by:

$$g_D(s) = \sum_{k=1}^{N_s - 1} \gamma_{b_k^s} + \mu \sum_{k=1}^{N_s} \ell_k^s \tag{6.19}$$

Like in the lazy approach, the cost-to-destination for the destination-attracted partial neutral fibre can be evaluated using the polyline $\left[p_1^s, ..., p_{N_s+1}^s\right]$ provided by destination-attracted linear program LP_s^D.

Definition 23: Destination-attracted heuristic h_D

Let s be a feasible routing plan in $\mathcal{S}$ and $\left[p_1^s, ..., p_{N_s+1}^s\right]$ the polyline provided by destination-attracted linear program LP_s^D which minimises the cost-to-destination. The *destination-attracted heuristic* for routing plan s is defined by:

$$h_D(s) = \begin{cases} \gamma_{min.}\left(o_{N_s}^s\right) + \mu \min_{p \in \mathcal{P}^{dest.}} \left\| \overrightarrow{p_{N_s+1}^s p} \right\| & \text{if } Term^s = true \\ 0 & \text{otherwise} \end{cases} \tag{6.20}$$

As illustrated on Figure 6.8, the partial neutral fibre gets as close as possible to the destination polyhedron $\mathcal{P}^{dest.}$ in order to minimise the cost-to-destination. Consequently, the cost-from-origin is predominant in the neutral fibre used to evaluate routing plan $s \in \mathcal{S}$.

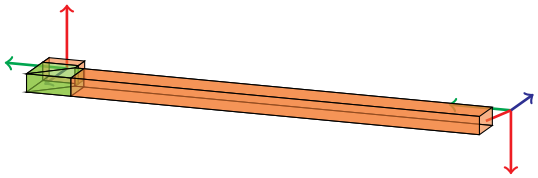
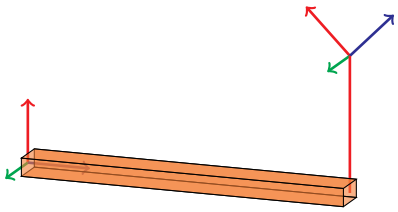
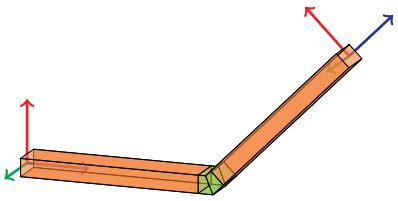


Figure 6.8 – Partial neutral fibre and polyline to the destination with the destination-attracted approach.

It can be easily proved that destination-attracted heuristic h_D is not admissible and then not consistent. For instance, Figure 6.9 illustrates a case where adding a bend $b \in B_{cat.}$ to a routing plan $s \in \mathcal{S}$ clearly reduces the linear cost of the complete neutral fibre in order to reach the destination polyhedron $\mathcal{P}^{dest.}$. In this case, if the added bend b has a zero cost $\gamma_b = 0$, then the estimation $g_D(s) + h_D(s)$ overestimates the cost of a destination plan which is a descendant of routing plan s.



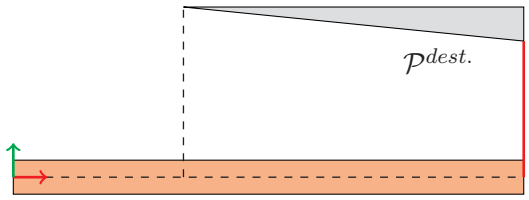
(a) Partial neutral fibre and polyline to the destination for a routing plan $s \in \mathcal{S}$.



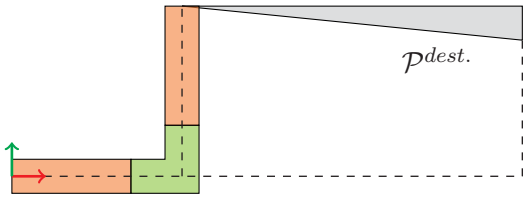
(b) Partial neutral fibre and polyline to the destination for plan $s \in \mathcal{S}$ extended with a bend $b \in B_{cat.}$.

Figure 6.9 – Example of non admissibility with the destination-attracted approach.

In the same way, Figure 6.10 shows an example where adding a zero-cost bend $b \in B_{cat.}$ to a routing plan $s \in \mathcal{S}$ decreases the cost-from-origin. So, the destination-attracted cost g_D is non-increasing when extending the routing plans. The example also illustrates that the estimation $f = g_D + h_D$ is non-increasing.



(a) Partial neutral fibre and polyline to the destination for a routing plan $s \in \mathcal{S}$.



(b) Partial neutral fibre and polyline to the destination for plan $s \in \mathcal{S}$ extended with a bend $b \in B_{cat.}$.

Figure 6.10 – Example of decreasing destination-attracted cost $g_D(s)$.

A*, WA* and more generally BFS approaches are optimal only if the evaluation function f is non-decreasing. Since $f_D = g_D + h_D$ is not, these algorithms will only provide suboptimal solutions to the FWRP. However, A* using $f_D(s) = g_D(s) + h_D(s)$ and WA* using $f_D(s) = g_D(s) + \epsilon h_D(s)$ will be compared with the versions using lazy heuristic h_L. Last, like in the lazy

approach, incomplete approaches like LSS-LRTA* with $f_D(s) = g_D(s) + h_D(s)$, and BrF-BS as well as SAHC with $h_D(s)$ are also studied because the more "aggressive" heuristic h_D may find solutions faster than heuristic h_L.

6.3 Experiments on the FWRP

Both introduced evaluation methods, the lazy and destination-attracted approaches, have been implemented in Java with the following ISAs: A*, WA*, Greedy BFS, SAHC, BrF-BS, and LSS-LRTA*. Remind that the implementation of these algorithms does not maintain any data structure of the already visited states since routing plans are built by forward chaining and so cannot be generated twice. The resolution of the linear programs has been performed using the Simplex solver of the Apache Commons Math library (version 3.6.1) [3]. Note that, in the results that follow, the search has been stopped as soon as a solution was found and the runtime has been limited to 1 minute.

First of all, each kind of algorithm (BFS, BS and HC) is studied separately in order to select the best parameters to solve the FWRP. Then, the best approaches as well as the MILP formulation presented in Chapter 5 are compared in Section 6.3.2.

6.3.1 Tuning the different Informed Search Algorithms

Best-First Search

WA* algorithm has been studied for the following weight ϵ-values: 1.1, 1.2, 1.5, 2, 3, 5 and 8. A*, which is WA* with $\epsilon = 1$, and Greedy BFS, which can be seen as WA* with $\epsilon = \infty$, are also tested. Figure 6.11 on page 86 and Figure 6.12 on page 87 present the results on the three instance sets (see Section 5.3.1 on page 63) using respectively the lazy and destination-attracted approaches.

First, it appears that all solutions found using A* and the lazy approach are optimal, which is consistent with the admissibility of lazy heuristic h_L, proved in Section 6.2.2. By contrast, the non-admissibility of the destination-attracted heuristic h_D is empirically validated by non-optimal solutions returned by A*. In both cases, all instances are solved since BFS approaches are complete.

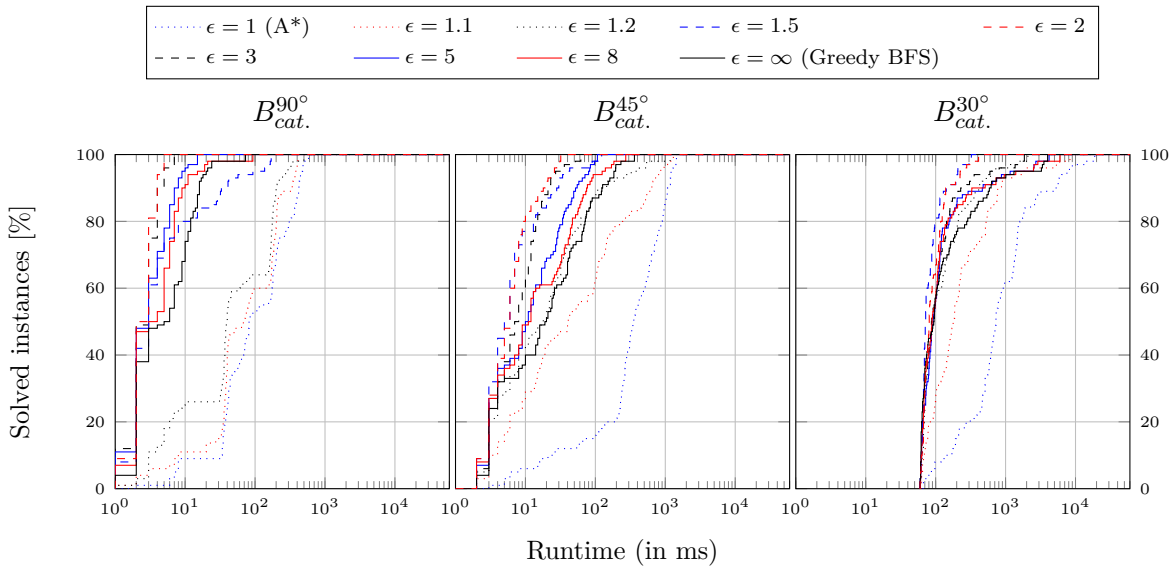
With the lazy approach $f_L = g_L + \epsilon h_L$, the gaps to the optimal solution cost decrease with weight ϵ, up to systematically reaching optimal solutions with $\epsilon = 1$ (see Figure 6.11c). The behaviour of the resolution time, which is correlated to the number of iterations performed, is divided into two parts: first, it decreases when weight ϵ rises, then it starts to increase after exceeding a threshold value of ϵ (see Figure 6.11b). Beyond this particular value, the heuristic contribution becomes misleading. It results that the best ϵ values are around this threshold since the runtime is low and the quality of the solution stays acceptable. Here, WA* provides the best performances on the three instance sets with $\epsilon = 1.2$, 1.5 and 2.

For the destination-attracted approach $f_D = g_D + \epsilon h_D$, the resolution runtime tends to decrease when weight ϵ rises, as illustrated on Figure 6.12b, but at the cost of a poorer solution quality (see Figure 6.12c). However, although there is no theoretical upper bound on the gap to the optimal cost when using this approach, it can be noticed that this gap grows slowly with weight ϵ and stays generally under 50%. So, this evaluation method can be competitive with the lazy approach by choosing $\epsilon = 2$, 3 or 5, which is a good trade-off between resolution speed and solution quality.

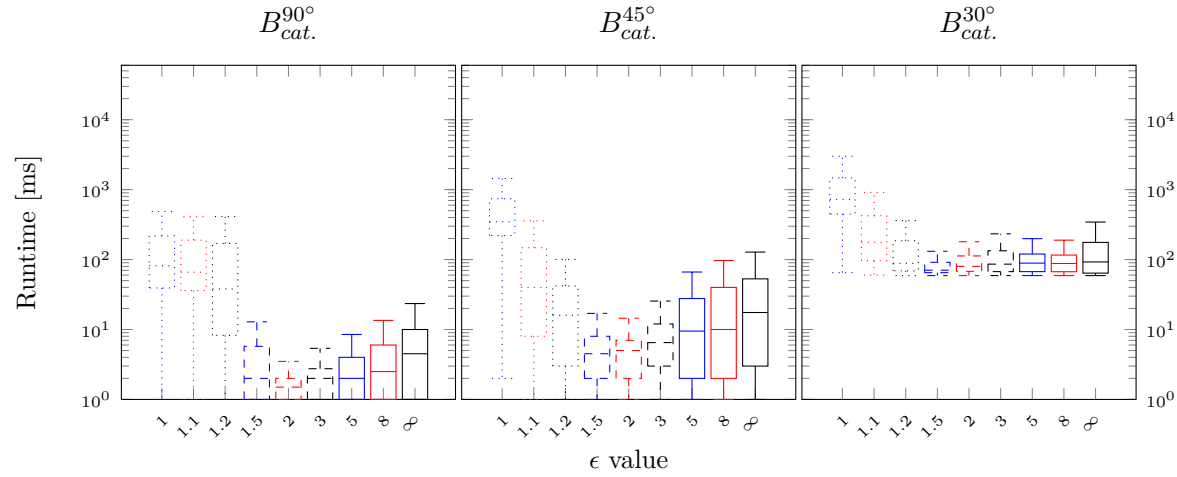
Furthermore, by comparing the number of iterations performed by both evaluation methods on Figure 6.13, it appears that the destination-attracted approach expands fewer routing plans and leads more directly to a solution. In particular, this can be verified on instances where

the origin and destination polyhedrons are distant and the bend costs are low. In these cases, the cost-to-destination becomes preponderant in the lazy evaluation, since the partial neutral fibre stays around the origin. Thus, the cost of routing plans from a same level is almost the same, which results in a high number of expanded plans. Nevertheless, the destination-attracted approach requires solving a linear program LP_s^D each time a plan $s \in \mathcal{S}$ is evaluated. By contrast, an optimal solution is known for lazy linear program LP_s^L when a bend is added (see Property 3). So, it is possible to avoid the resolution of linear program LP_s^L using the simplex algorithm in most cases and, consequently, to perform many more iterations by time unit. This is the reason why the lazy approach can be faster even if it expands more routing plans.

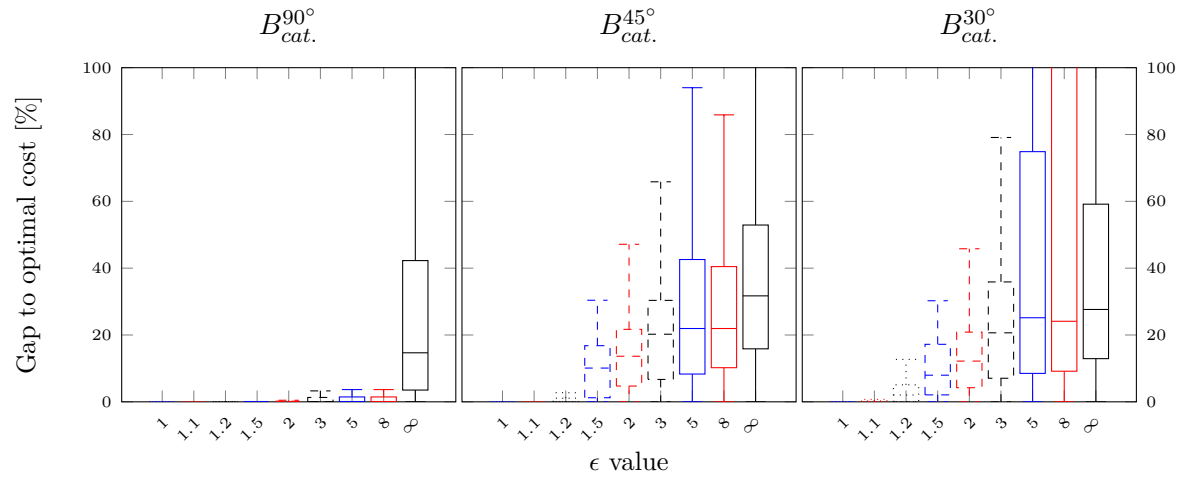
Both evaluation methods are compared with their best weight values on Figure 6.14 on page 89. The lazy approach provides a solution quality comparable to the destination-attracted approach, but faster than the latter. Moreover, the biggest advantage of the lazy evaluation $f_L = g_L + \epsilon h_L$ is that the gap to the optimal cost is upper bounded because of the consistency of heuristic h_L. Finally, WA* with $\epsilon = 1.5$ and the lazy approach seems to be the most efficient BFS method.



(a) Evolution of success rates.

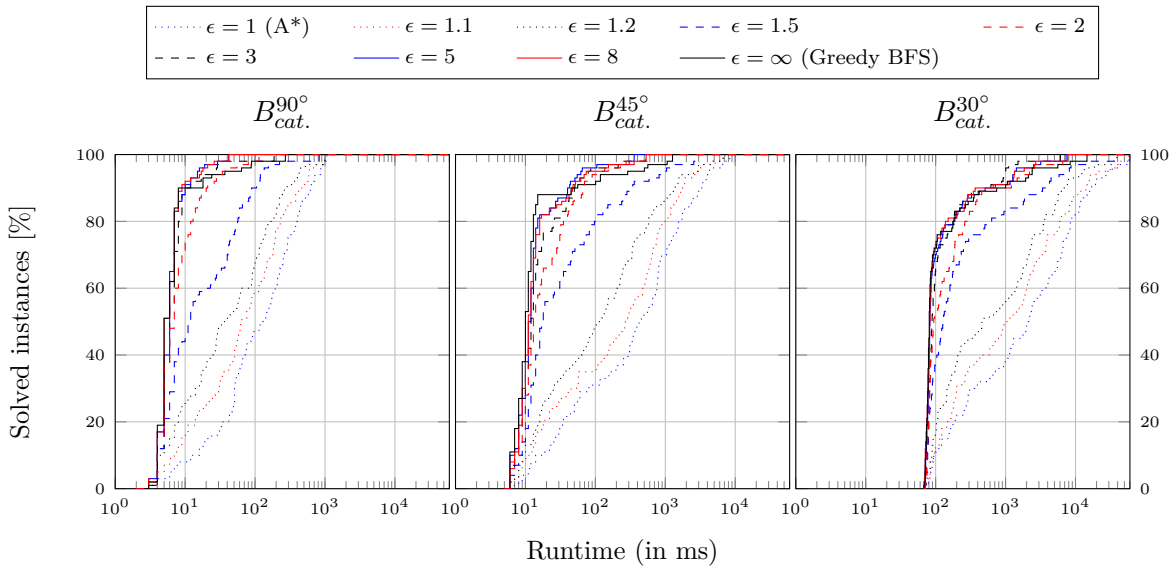


(b) Runtimes.

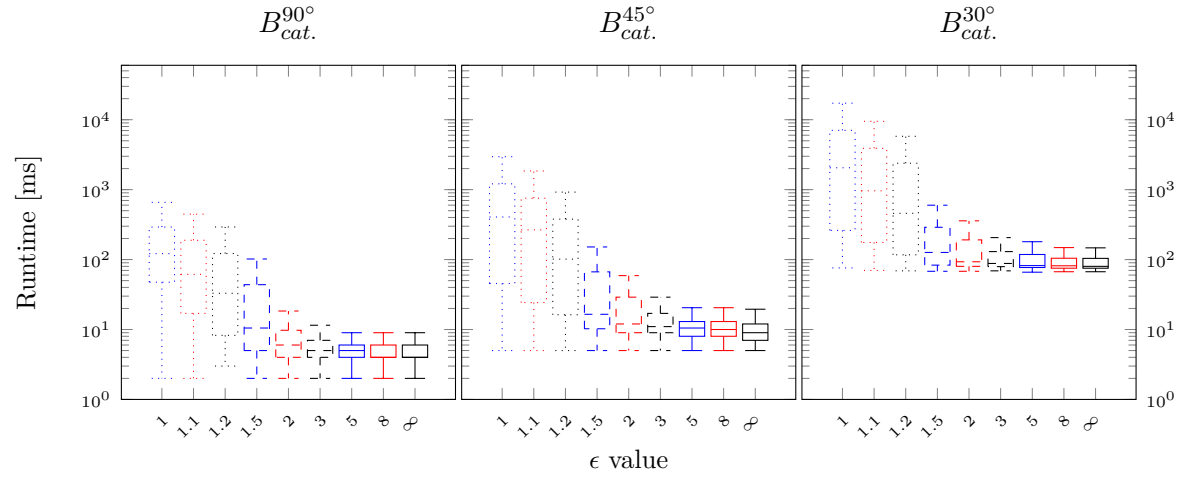


(c) Gaps to the optimal cost.

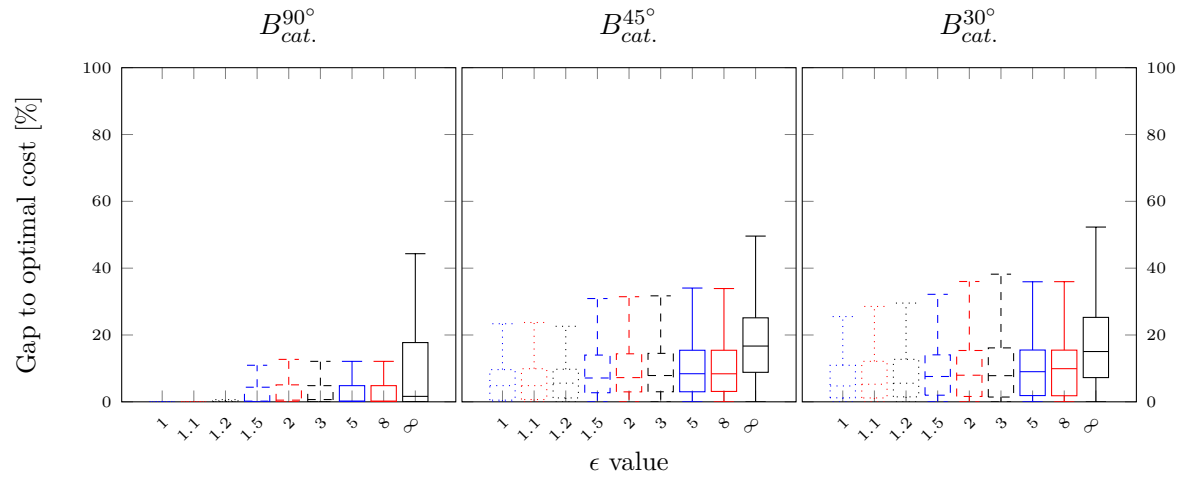
Figure 6.11 – Results with A*/WA* and the lazy approach.



(a) Evolution of success rates.

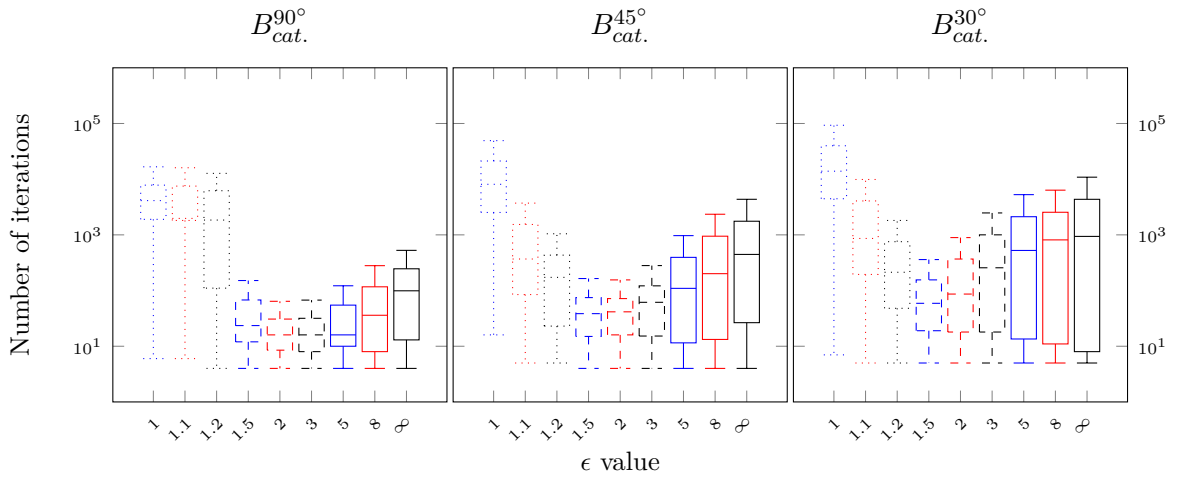


(b) Runtimes.

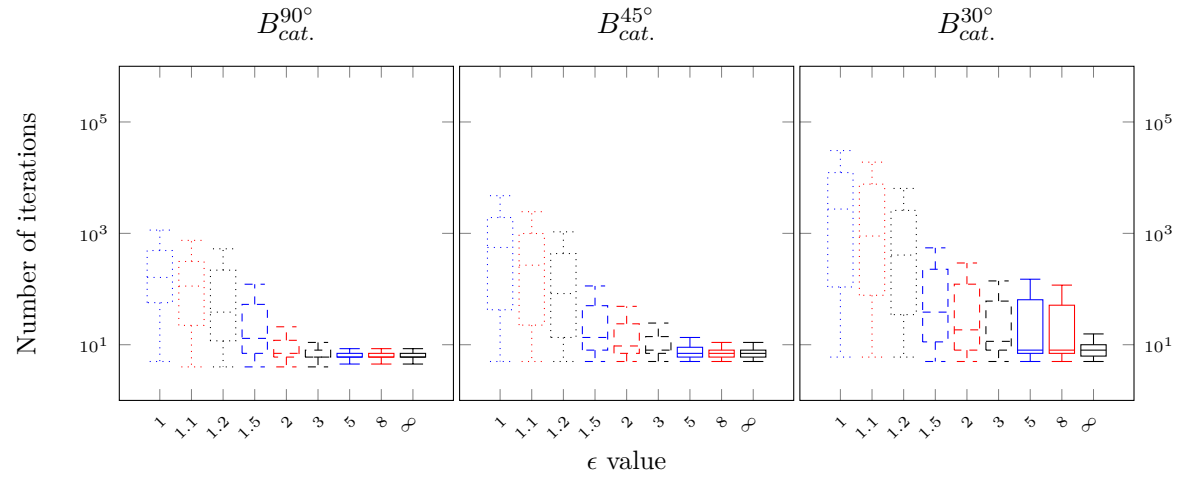


(c) Gaps to the optimal cost.

Figure 6.12 – Results with A*/WA* and the destination-attracted approach.

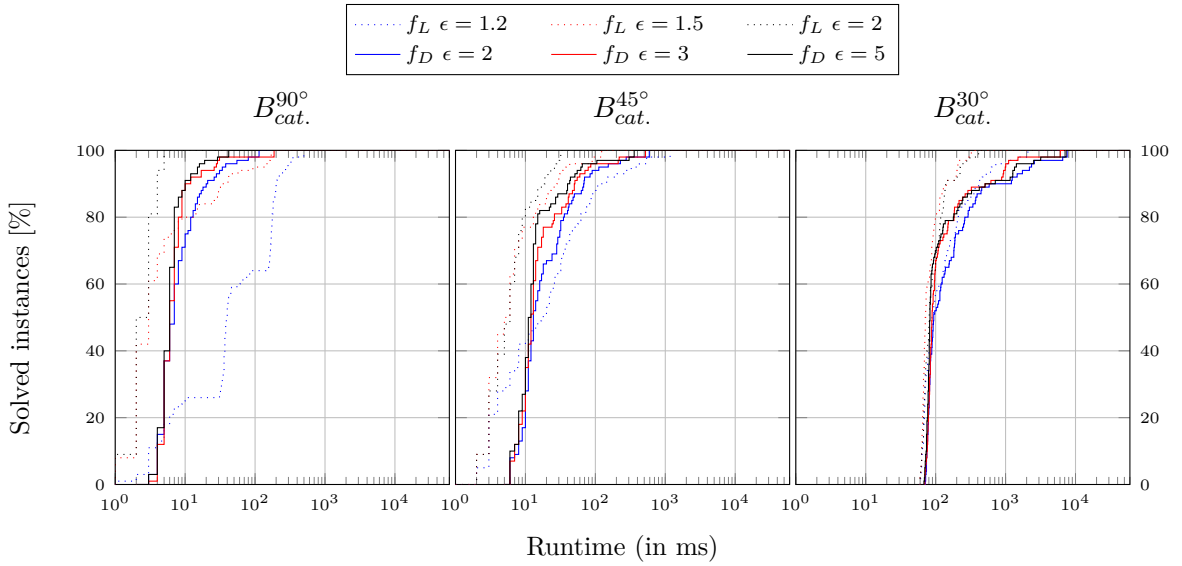


(a) Lazy approach.

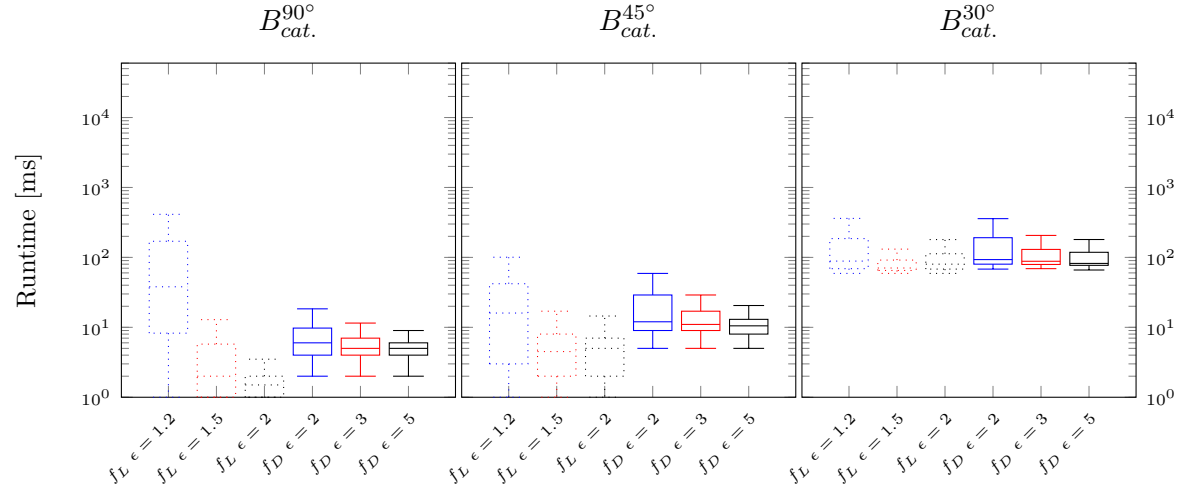


(b) Destination-attracted approach.

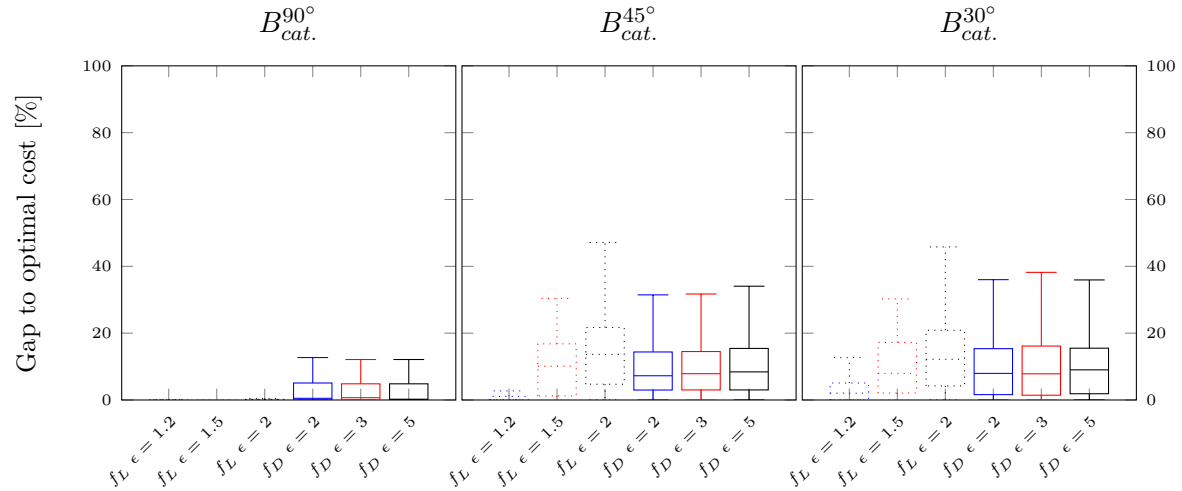
Figure 6.13 – Comparison of the number of iterations performed with A*/WA* .



(a) Evolution of success rates.



(b) Runtimes.



(c) Gaps to the optimal cost.

Figure 6.14 – Best results with A*/WA*.

Beam Search

BrF-BS has been studied for the following beamwidth W-values: 2, 5, 10, 20, 50, 100, 200 and 500. SAHC, which can be seen as BrF-BS with $W = 1$, is also tested. Figure 6.15 on the next page and Figure 6.16 on page 92 present the results on the three instance sets using respectively the lazy and destination-attracted heuristics h_L and h_D.

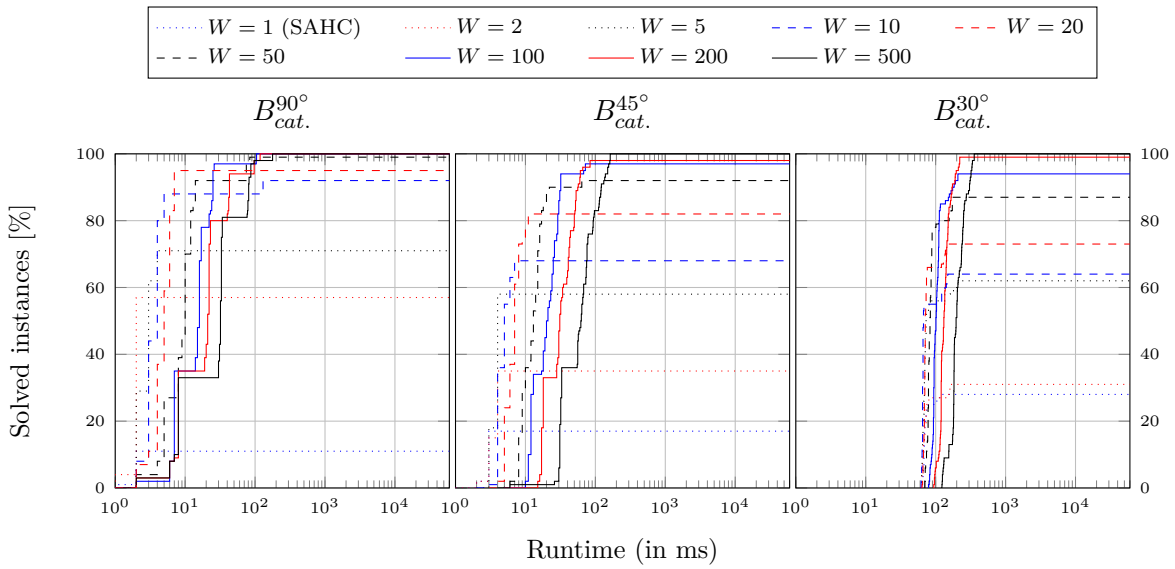
For both heuristics, resolution runtimes grow with beamwidth W. Nevertheless, a large beamwidth is required with the lazy heuristic h_L in order to ensure an acceptable success rate. Typically, only W values higher than 100 allow solving more than 90% of instances with the largest catalogue $B^{30^\circ}_{cat.}$ (see Figure 6.15a). In comparison, the destination-attracted heuristic h_D reaches equivalent success rates with a thinner beamwidth since h_D leads more directly to a solution, as illustrated on Figure 6.16a. For instance, a 10-beamwidth BrF-BS already solves more than 90% of instances using $B^{30^\circ}_{cat.}$. In particular, it can be noticed that even SAHC has a high resolution rate around 75% on $B^{30^\circ}_{cat.}$ instances within less than 200 milliseconds. Generally, using a thinner beamwidth makes BrF-BS faster, but the lazy heuristic h_L compensates its disadvantage by avoiding the resolution of the lazy linear program for each generated routing plan and stays more efficient than h_D, even with larger beamwidths (see Figure 6.15b and Figure 6.16b).

It can also be seen that the solution quality rises with beamwidth W for both heuristics. However, the destination-attracted heuristic h_D provides reasonable gaps to the optimal solution cost even with small beamwidths (see Figure 6.16c), while heuristic h_L requires a minimal beamwidth to ensure gaps lower than 50% (see Figure 6.15c). Moreover, when beamwidth W is large, h_D still gives slightly better solutions than h_L. This may be partially explained by the fact that lazy heuristic h_L can be misleading with BrF-BS, which only considers the remaining cost $h(s)$ to reach a destination plan. Indeed, lazy heuristic h_L enforces the partial neutral fibre to be close to the origin polyhedron $\mathcal{P}^{ori.}$. It results that, in cases where bend costs are low and bend catalogue $B_{cat.}$ contains a twist, there are configurations in which applying a twist reducing the distance to destination polyhedron $\mathcal{P}^{dest.}$ is more promising than applying another bend, even if the latter makes it possible to reach the destination at the next step. Therefore, solutions found with lazy heuristic h_L may contain unnecessary twists. In comparison, destination-attracted heuristic h_D favours routing plans that get close to destination polyhedron $\mathcal{P}^{dest.}$, so applying a twist does not have an impact on the remaining distance to $\mathcal{P}^{dest.}$. As a result, the quality of solutions found by h_D is generally better.

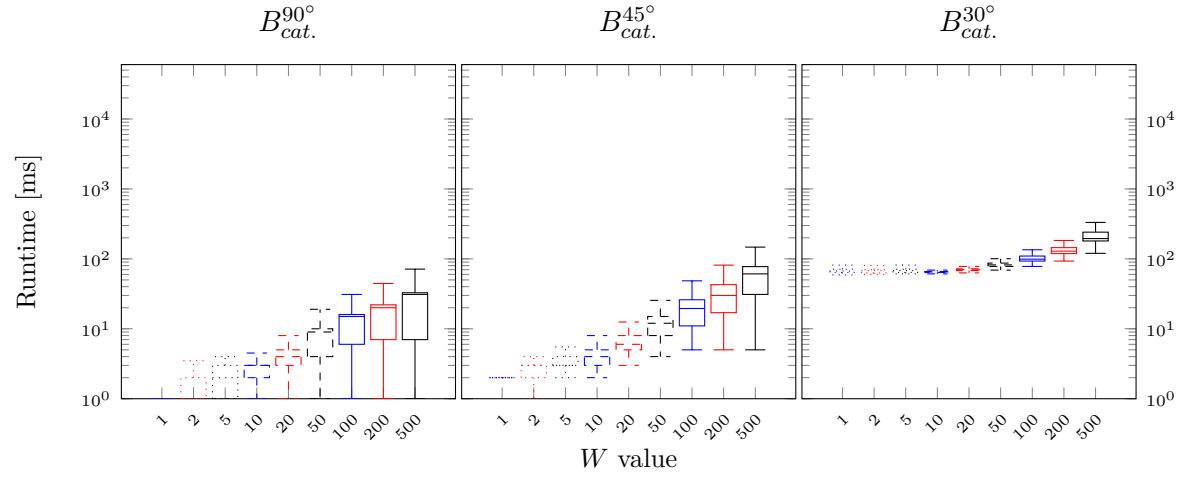
The smallest W values that ensure acceptable resolution rates are 100, 200 and 500 for heuristic h_L, and 50, 100 and 200 for heuristic h_D. By comparing them (see Figure 6.17 on page 93), it appears that the lazy heuristic h_L is clearly faster than h_D when using BrF-BS. Since the solution quality improvement brought by the destination-attracted approach is negligible, it is preferable to use heuristic h_L. Finally, beamwidth value $W = 500$ and heuristic h_L is the most efficient tuning of BrF-BS to solve all instances.

Hill-Climbing

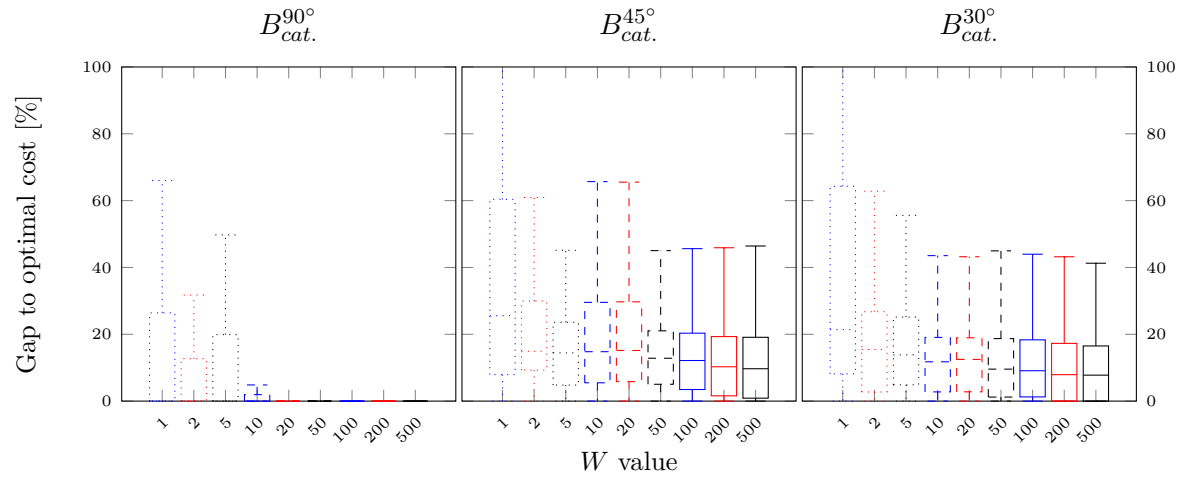
Figure 6.18a and Figure 6.18b show the evolution of success rates with LSS-LRTA* using respectively lazy and destination-attracted evaluations. Values 1, 2, 5, 10, 20, 50, 100, 200 and 500 have been tested for lookahead L. If this algorithm provides solutions fast in less than one second, it appears that it has a very poor success rate with both evaluation methods. Indeed, the success rate is capped to 45% on the largest catalogue $B^{30^\circ}_{cat.}$ with the lazy approach. Moreover, although this rate seems to increase with lookahead L when the destination-attracted evaluation is used, LSS-LRTA* is outperformed by A* with $f_L = g_L + h_L$. Consequently, LSS-LRTA* was not selected among the best candidate algorithms.



(a) Evolution of success rates.

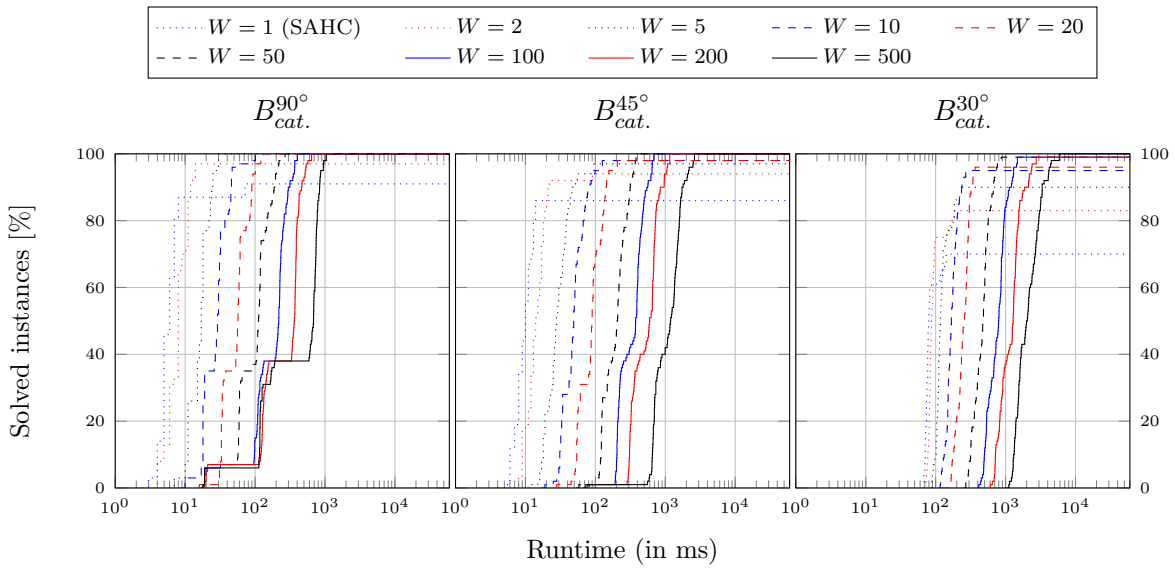


(b) Runtimes.

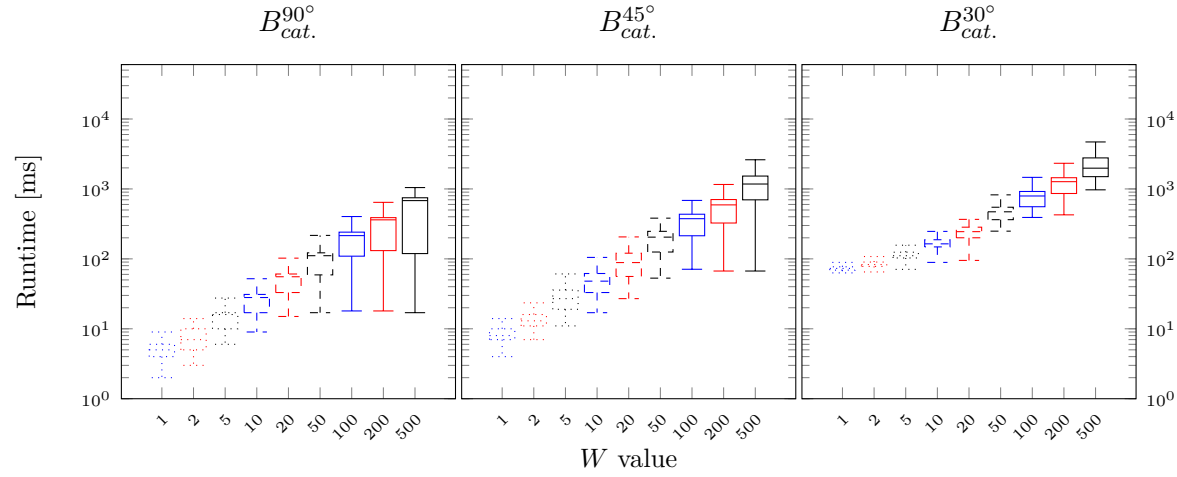


(c) Gaps to the optimal cost.

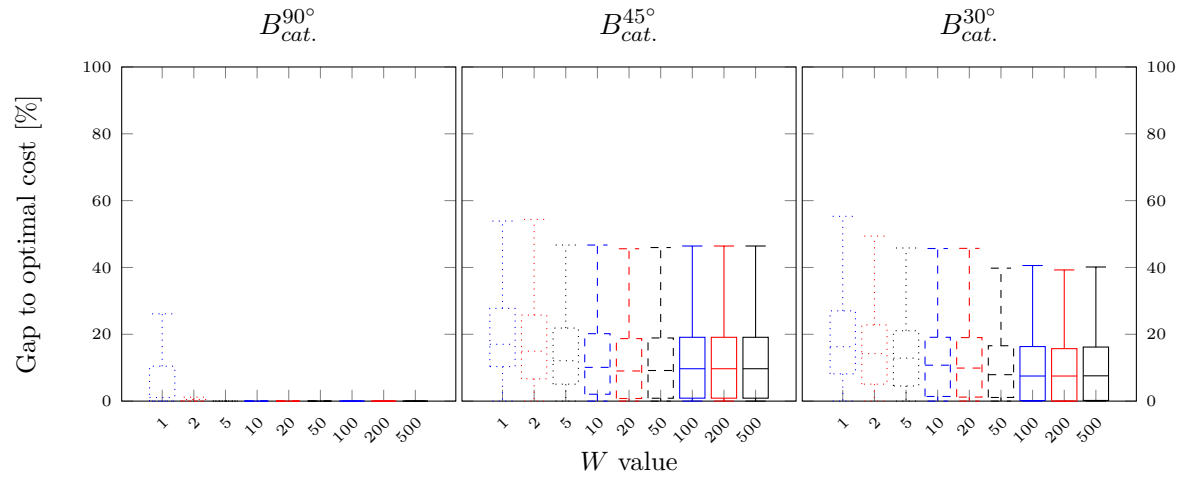
Figure 6.15 – Results with SAHC/BrF-BS and the lazy approach.



(a) Evolution of success rates.

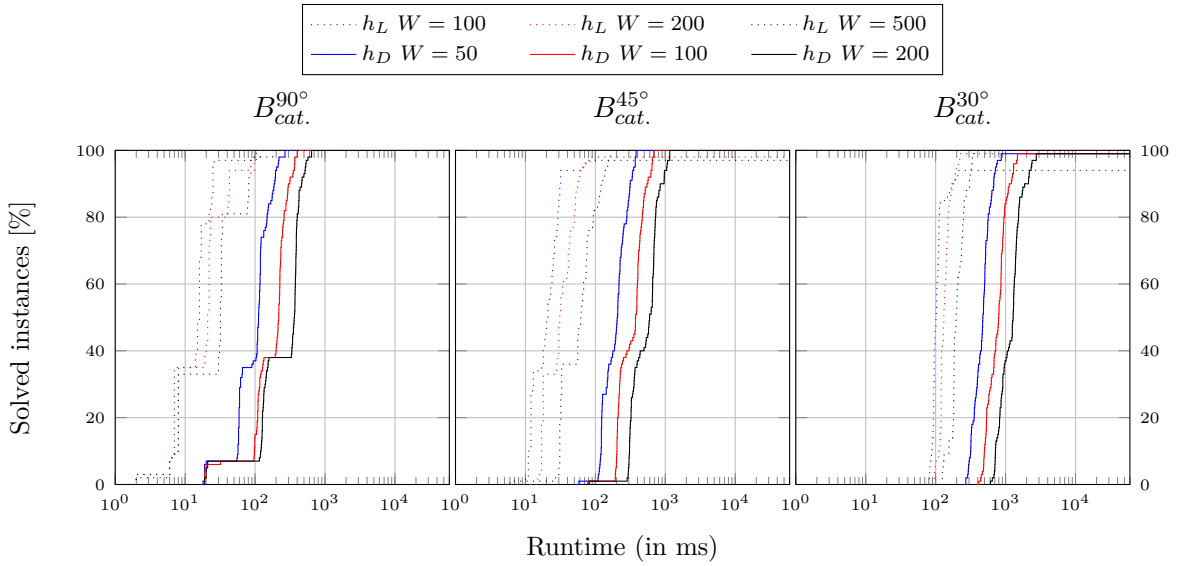


(b) Runtimes.

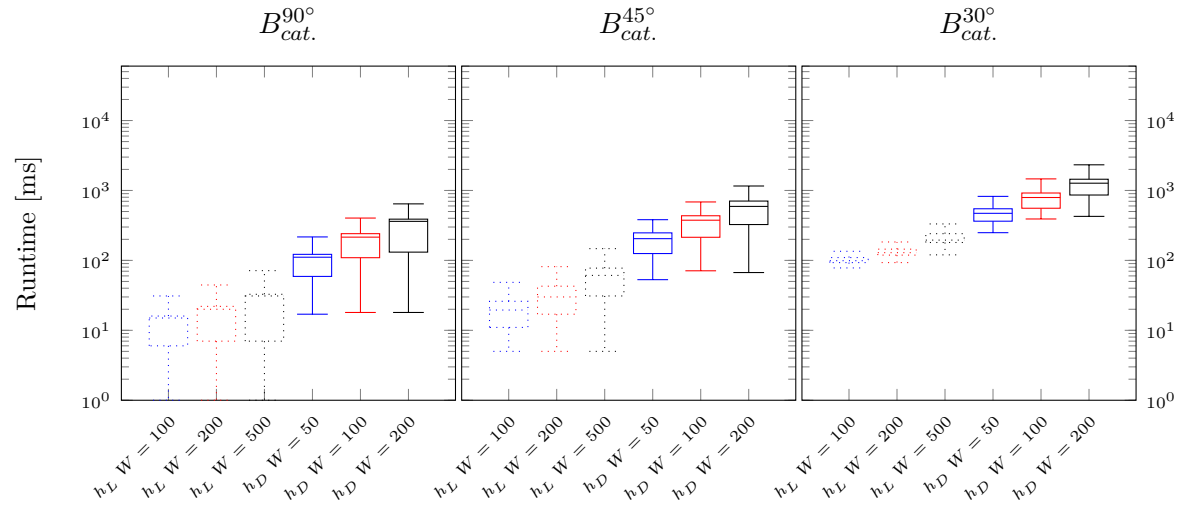


(c) Gaps to the optimal cost.

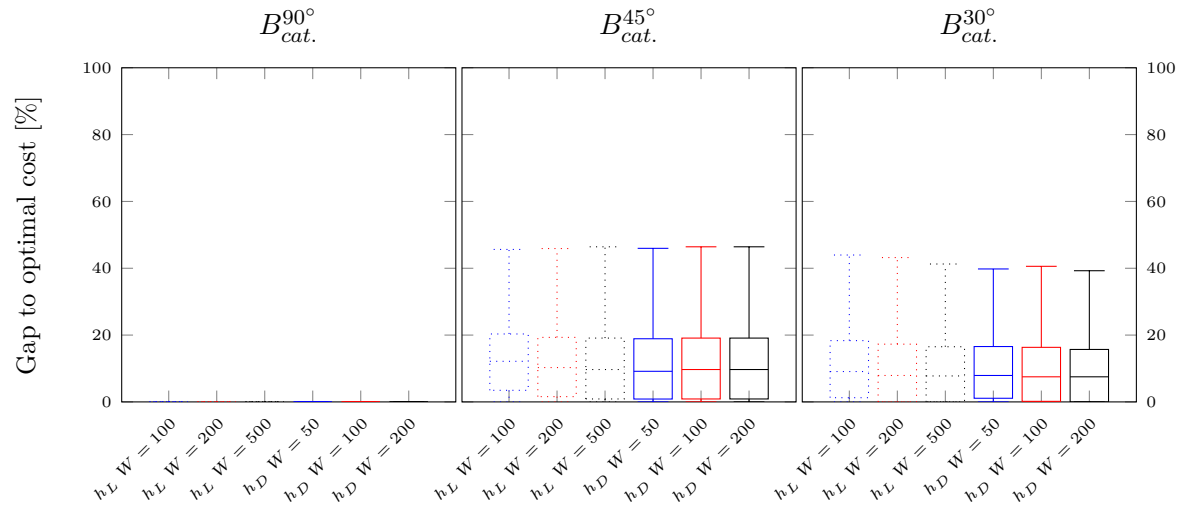
Figure 6.16 – Results with SAHC/BrF-BS and the destination-attracted approach.



(a) Evolution of success rates.

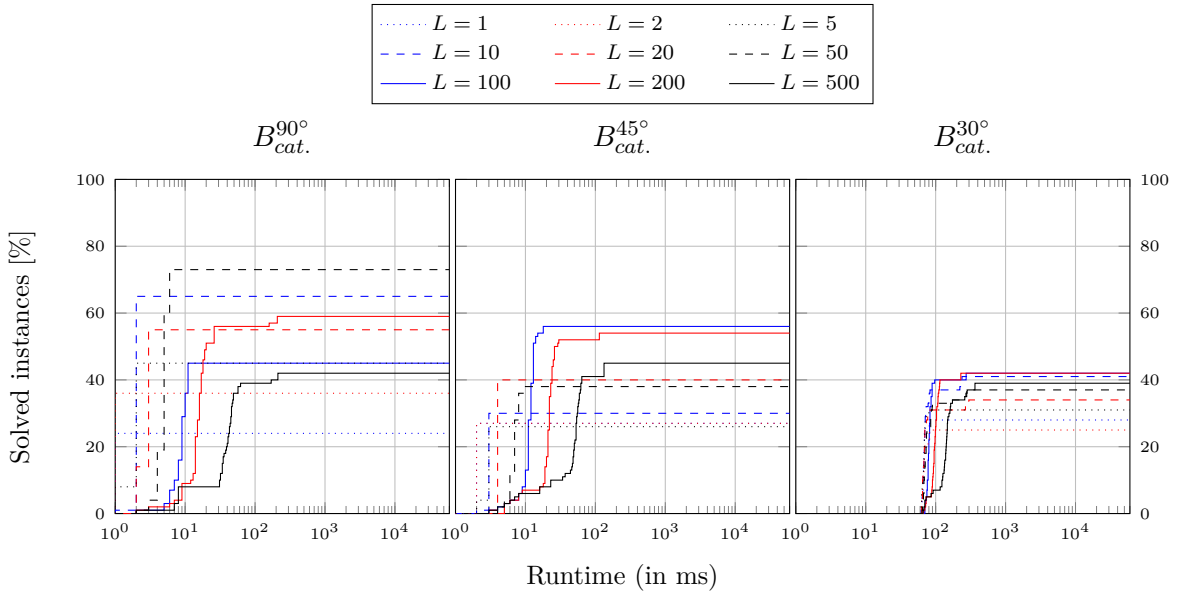


(b) Runtimes.

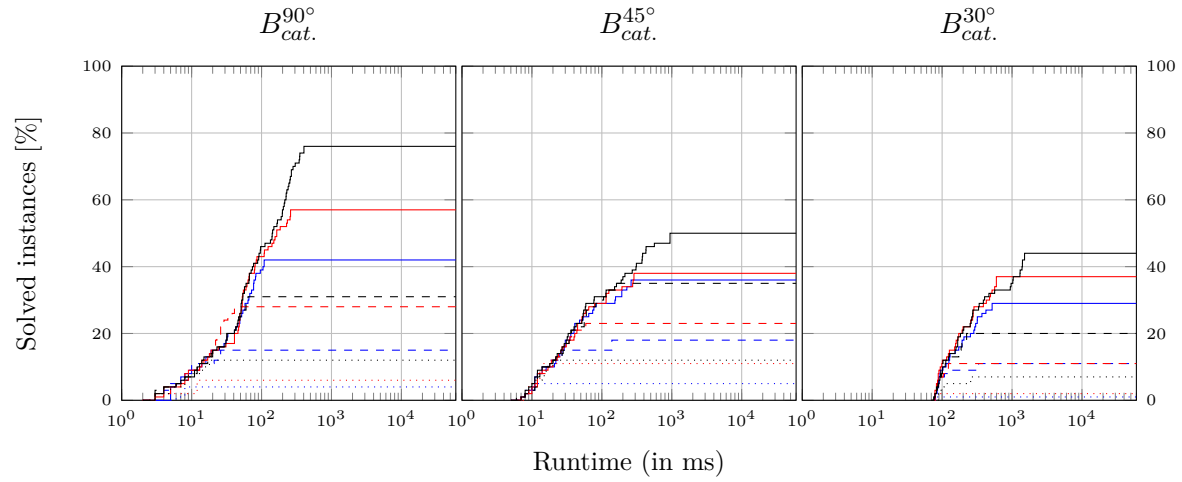


(c) Gaps to the optimal cost.

Figure 6.17 – Best results with SAHC/BrF-BS.



(a) Lazy approach.



(b) Destination-attracted approach.

Figure 6.18 – Evolution of success rates with LSS-LRTA*.

6.3.2 Comparison of the best approaches

As previously shown, the best explored ISAs to solve the FWRP are:

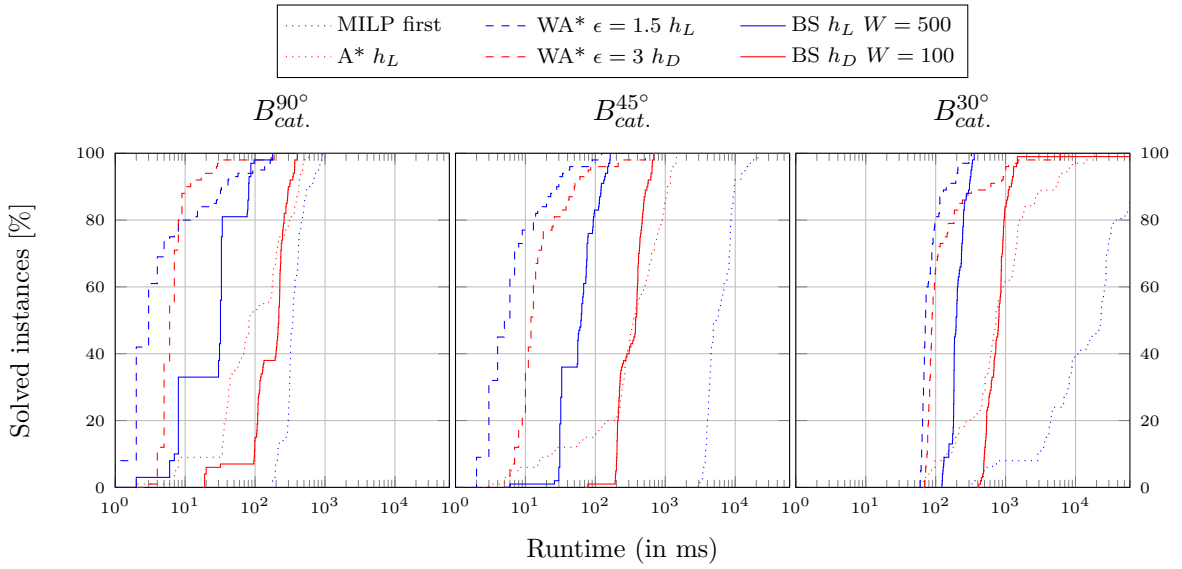
- A* using the lazy evaluation $f_L = g_L + h_L$;
- WA* using $\epsilon = 1.5$ and the lazy evaluation $f_L = g_L + \epsilon h_L$;
- WA* using $\epsilon = 3$ and the destination-attracted evaluation $f_D = g_D + \epsilon h_D$;
- BrF-BS using $W = 500$ and the lazy heuristic h_L;
- BrF-BS using $W = 100$ and the destination-attracted heuristic h_D.

These algorithms, as well as the resolution of the MILP model stopped as soon as a first solution is found, have been compared on the three instance sets using the maximal numbers of segments $N_S = 6$ and $N_S = 10$. Results are presented on Figure 6.19 and Figure 6.20 respectively.

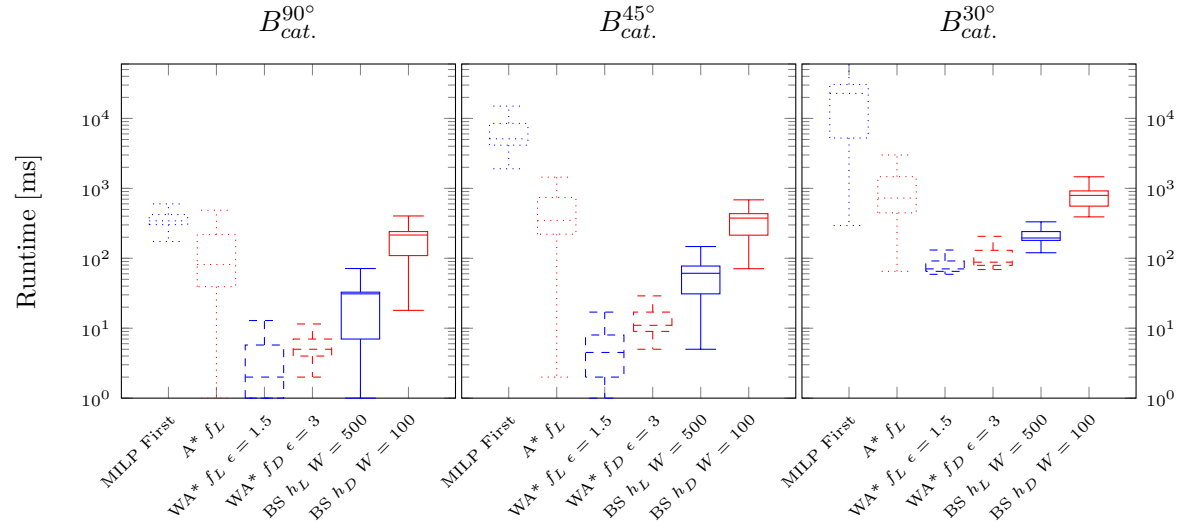
It appears that the MILP formulation is outperformed by the ISAs. Indeed, WA* and BrF-BS provide solutions within runtimes lower by several orders of magnitude compared to MILP stopped when reaching the first solution, as shown on Figure 6.20b. Moreover, the quality of these solutions is significantly better with ISAs (see Figure 6.19c) and is acceptable for designers as illustrated on Figure 6.21. When the maximal number of segments N_S is low, even A* reaches the optimal solution before MILP encounters its first solution, as illustrated on Figure 6.19a. Furthermore, the MILP approach is very sensitive to the number of segments N_S because of the combinatorial explosion due to a higher number of possible orientation changes. For instance, it did not provide any solution within a minute on bend catalogue $B^{30°}_{cat.}$ with $N_S = 10$ (see Figure 6.20a). In comparison, WA* and BrF-BS are relatively robust when N_S rises. This can be explained by the fact that increasing N_S does not impact the depth of destination routing plans which are reachable with a lower value of N_S. To make the MILP formulation less sensitive to N_S, it would be possible to solve a sequence of MILP models without introducing neutral bends in the space of candidate orientations. This approach has not been explored.

Among ISAs, the best trade-off between resolution speed and solution quality is reached using WA* with $\epsilon = 1.5$ and the lazy evaluation $f_L = g_L + \epsilon h_L$. BrF-BS finds more often the optimal solution, but its runtime is longer and the gap to the optimality is larger in the worst cases. Moreover, BrF-BS does not provide any guarantee on the solution quality. In contrast, WA* with the lazy evaluation ensures that the first solution found costs at most ϵ times the optimal cost because of the consistency of heuristic h_L. Last, if A* always finds the optimal solution, its resolution speed is not competitive since it explores many more routing plans. Of course, one of the most important advantages of BFS approaches over BrF-BS is their completeness. Whatever the ϵ value, WA* will find a solution if there exists one. Such an assertion does not hold for BrF-BS which can require a minimal beamwidth W to reach a solution.

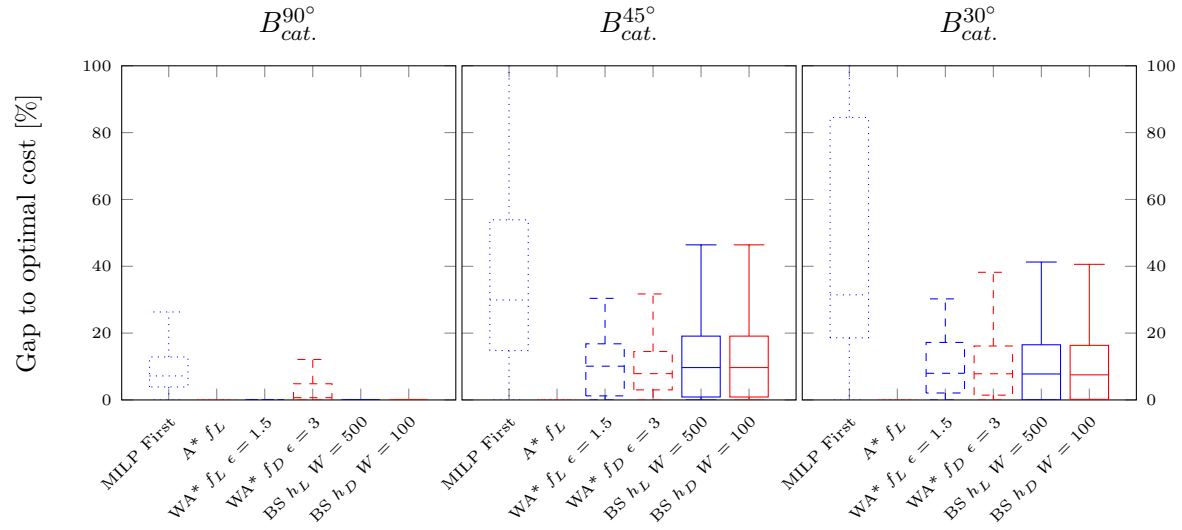
With WA*, it is possible to offer the ϵ value as a tuning parameter to the routing algorithm user in order to adjust the trade-off between resolution speed and optimality. To do so, the value domain of ϵ must be $[1, 2]$ since, as illustrated on Figure 6.11, runtime raises beyond $\epsilon = 2$. An ϵ value closer to 1 improves the optimality of the solution, while an ϵ value closer to 2 reduces the resolution speed.



(a) Evolution of success rates.

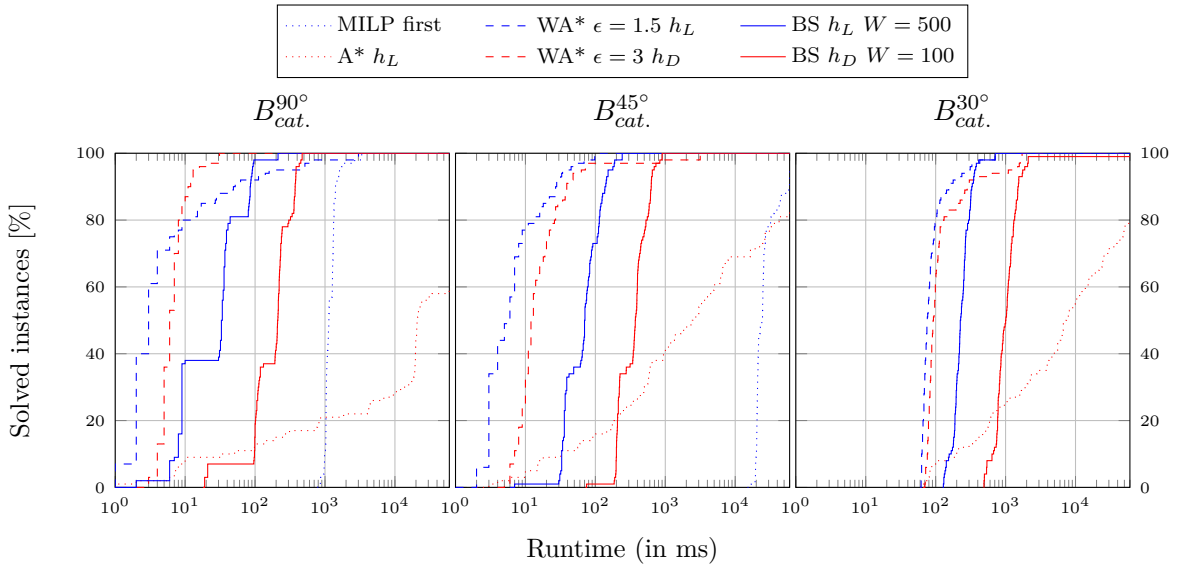


(b) Runtimes.

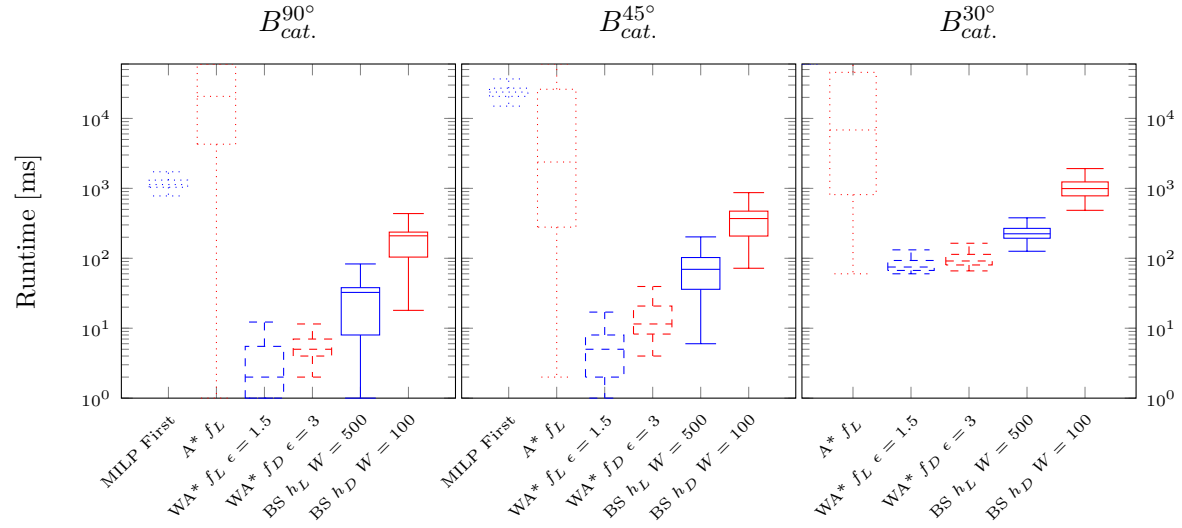


(c) Gaps to the optimal cost.

Figure 6.19 – Results with the best approaches with $N_S = 6$.



(a) Evolution of success rates.



(b) Runtimes.

Figure 6.20 – Results with the best approaches with $N_S = 10$.

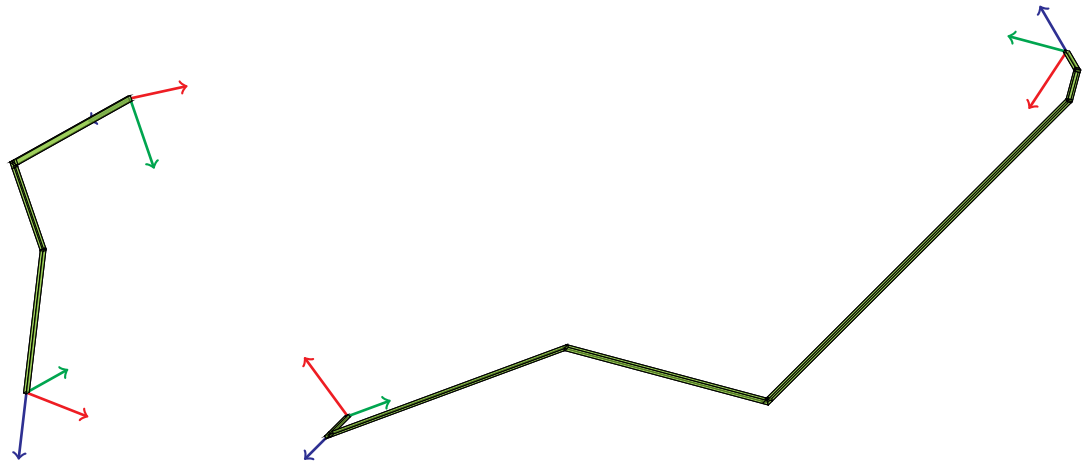


Figure 6.21 – Examples of routed waveguides with WA*.

Chapter 7

Conclusion on the FWRP

7.1 Contributions

In Part II, two formulations have been proposed in order to solve the Free Waveguide Routing Problem (FWRP): a first one using Mixed Integer Linear Programming (MILP) and another one as a Search Problem (SP). Both deal with potentially non-orthogonal bends defined in a catalogue and with unsymmetrical cross-sections, which is necessary to provide solutions that satisfy all the constraints for Waveguide Routing in the Radio-Frequency Harness of a telecommunication satellite. The formulations introduced are based on the enumeration of possible orientations for the waveguide segments. This is feasible since the number of candidate orientations is not that large thanks to the global attachability constraint. Then, both approaches use Linear Programming (LP) to route the waveguide in a continuous domain, considering local constraints on each straight section. It results that the solutions generated are realistic and can be used without modification by designers in cases where there is no conflict with other components or existing waveguides. The main ideas of these methods have been presented in publications at the ICORES 2020 [88] and ROADEF 2020 [89] conferences.

The resolution with the MILP formulation turned out to be too time-consuming, even if the method is attractive because it guarantees to provide optimal solutions. Indeed, when the size of the bend catalogue or the maximal number of segments increases, the runtime required to reach a first solution rises exponentially. It results that, on instances encountered in an industrial context, the poor performances of the MILP approach do not enable performing fast and numerous iterations during the design phase.

By contrast, the Informed Search Algorithms (ISAs), that can be used thanks to the SP formulation, are able to quickly provide solutions. These algorithms take full advantage of heuristics evaluating routing plans that define partially routed neutral fibres. Two evaluation methods have been introduced: the lazy approach, based on a linear program minimising the cost of the neutral fibre part that is already constrained by the routing plan, and the destination-attracted approach, which uses a linear program minimising the remaining distance to the destination polyhedron. Various types of ISAs have been experimented with these evaluation methods, including A*, WA*, Breadth-First Beam Search, and LSS-LRTA*. Even if the destination-attracted approach leads more directly to a solution in terms of number of iterations performed, the lazy approach takes full advantage of straightforward optimal solutions to save many simplex resolutions and reduce iteration duration. Moreover, the lazy heuristic has been proved to be consistent, which can provide guarantees on the solution quality depending on the type of ISA used. In particular, A* with the lazy heuristic finds the optimal solutions even before MILP encounters its first solutions. More generally, most ISAs clearly outperform the MILP formulation. Among them, WA* with the lazy approach appears to be the most efficient method.

Indeed, this algorithm is complete and able to solve industrial instances within a few hundred milliseconds while ensuring an upper bound on the gap to optimality. Furthermore, it offers the possibility to make a trade-off between resolution speed and solution quality by tuning the heuristic weight.

As a result of these good performances, the optimisation method designed for the FWRP has been implemented and integrated in an Airbus DS software suite and provided to waveguide designers. Even if the proposed routing algorithm does not take conflicts with other units into account, it is particularly appreciated because its resolution speed enables fast and numerous iterations during the design of a waveguide. Moreover, it is possible to deal with conflicts by defining routes from a higher level, via intermediate waypoints, rather than defining in a detailed manner the sequence of bends and straight sections used to avoid a conflict. This approach has been applied during the response to the tender phase of an actual telecommunication satellite. Classically, the RF-harness routing can last up to 18 weeks for a single designer. The waveguide routing algorithm in free space with the intermediate waypoint approach is estimated to reduce this duration by half. Furthermore, the usage of a bend catalogue constrains designers to use classic bend angles, which favours the standardisation of waveguides in the RF-harness.

7.2 Perspectives

The different proposed approaches to solve the FWRP could be further improved in the following ways. First of all, the introduced LP models suffer from being permissive to self-conflicting waveguide solutions. Indeed, there is no constraint to avoid a waveguide segment to intersect with another one. So, in cases where the distance between the origin and destination polyhedrons is small but greater than the tolerance, a solution using a loop, as illustrated on Figure 7.1, is valid with the current LP models. Note that, in practice, such self-intersecting solutions are hardly ever produced.

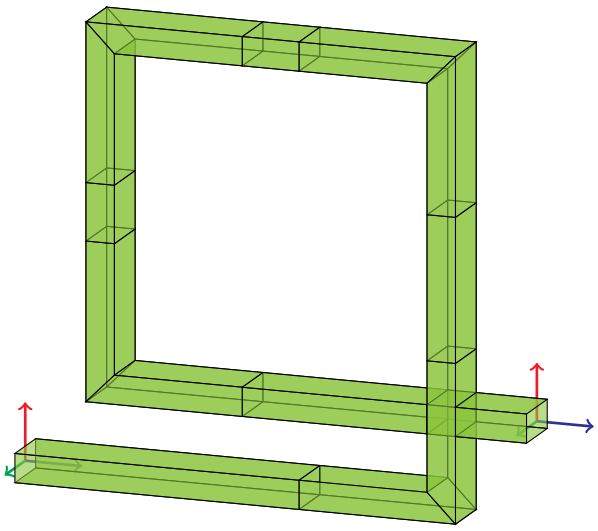


Figure 7.1 – Example of self-conflicting waveguide.

Then, if it is currently outperformed by ISAs, the MILP formulation may be more competitive when using a more powerful solver, like CPLEX or GUROBI. In addition, the approach could be further improved thanks to advanced LP techniques. A column generation formulation may be used considering the bend sequence sub-problem to enumerate bend combinations as columns for a master problem which would deal with the 3D-position sub-problem.

In the same way, the destination-attracted approach may become more efficient than the lazy one by reducing evaluation duration. To do so, the linear program for a routing plan could be warm-started by reusing the result basis of the simplex resolution of its parent's linear program.

Moreover, the simple Breadth-First Beam Search method, that has been used in this thesis, could be replaced by more advanced Beam Search algorithms, like BULB [29] or Beam-Stack Search [103] which are complete algorithms. More generally, the SP formulation as well as the proposed evaluation methods allow using various kinds of ISAs.

Of course, the biggest avenue to explore remains the extension of the proposed approaches to deal with spatial constraints and obstacles. This is the subject of Part III.

Part III

Waveguide Routing in Constrained Space

Chapter 8

Constrained Waveguide Routing Problem

This chapter presents the Constrained Waveguide Routing Problem (CWRP) which is an optimisation problem consisting in routing a single waveguide within a restricted three-dimensional space that may contain obstacles. It is an extension of the FWRP with space constraints. First, the modelling of a routing space is presented in Section 8.1 as well as a methodology to build it in practical applications. Then, the formalisation of the CWRP is introduced in Section 8.2 that describes the additional constraints the waveguide must satisfy.

8.1 Modelling a routing space

In order to take obstacles and other space constraints into account in the CWRP, it is necessary to model the routing space in which the waveguide must be contained. In the case of the RF-harness, waveguides must be fixed using brackets on walls and panels which make up the structure of the telecommunication satellite, as illustrated on Figure 8.1. To comply with the bracket maximal height, the routing space is located in the immediate vicinity of these panels. It results that the available space is naturally divided into several areas. Of course, waveguides must also avoid the other components that are already placed and fixed on the satellite panels, which makes the routing space even more complex.

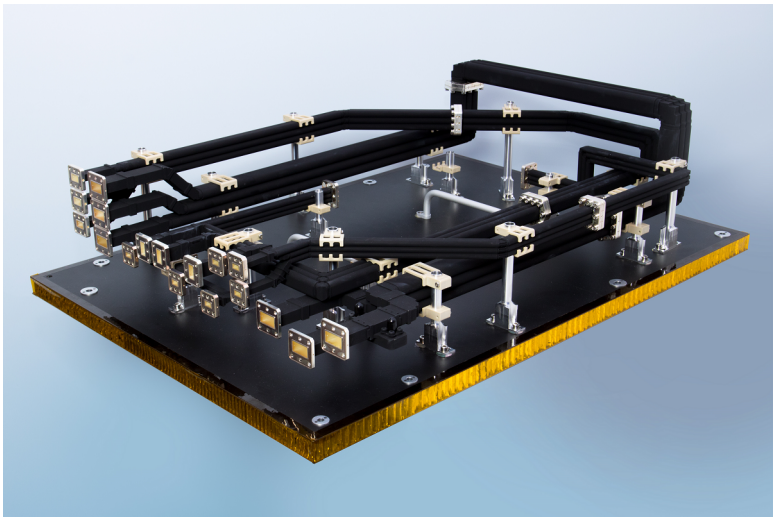


Figure 8.1 – Waveguides fixed on a panel.

8.1.1 Simplified model

The routing space can be described as a set of convex polyhedrons, called *traversable cells*, in which the neutral fibre $\mathcal{F}_\pi$ of waveguide π must be contained. By definition, these polyhedrons do not overlap any obstacle in order to avoid conflicts with other components of the satellite payload. The main purpose of the convexity assumption is to ensure that Linear Programming will still be applicable in the extension of the formulations already proposed to solve the FWRP (see Part II).

Definition 24: Traversable cell

A *traversable cell* c of the routing space describes a non-empty bounded convex polyhedron $\mathcal{P}_c$ in which the neutral fibre $\mathcal{F}_\pi$ of waveguide π can be routed (see Figure 8.2).
The set of traversable cells is referred to as $\mathcal{C}$.

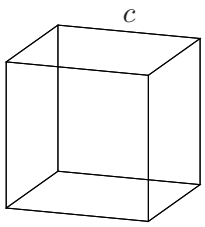


Figure 8.2 – A traversable cell $c \in \mathcal{C}$.

Definition 25: Crossable interface

Let c and c' be two traversable cells in $\mathcal{C}$ which overlap each other, that means $\mathcal{P}_c \cap \mathcal{P}_{c'} \neq \varnothing$. The *crossable interface* i from cell c to c' describes the non-empty bounded convex polyhedron $\mathcal{P}_i = \mathcal{P}_c \cap \mathcal{P}_{c'}$ that the neutral fibre $\mathcal{F}_\pi$ of waveguide π can cross to reach c' from c (see Figure 8.3).
The set of crossable interfaces is referred to as $\mathcal{I}$. Moreover, in what follows, the origin and destination traversable cells of a crossable interface $i \in \mathcal{I}$ are respectively referred to as c_i^- and c_i^+.

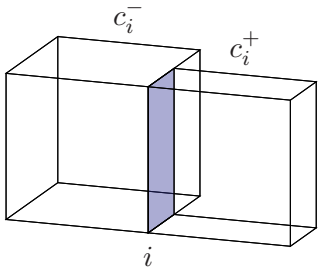


Figure 8.3 – A crossable interface $i \in \mathcal{I}$ from traversable cell $c_i^- \in \mathcal{C}$ to $c_i^+ \in \mathcal{C}$.

Since traversable cells are convex, it results that crossable interfaces, which are intersections of cells, are also convex. Futhermore, by definition, for each crossable interface $i \in \mathcal{I}$ from traversable cell c_i^- to c_i^+, there exists a reverse crossable interface referred to as $reverse\,(i) \in \mathcal{I}$ that goes out of cell c_i^+ and reaches cell c_i^- and is defined by the same polyhedron $\mathcal{P}_i$.

It is assumed that traversable cells of the routing space have been built such that all intersections between two cells are either empty or 3D-polygons.

Assumption 2: Polygonal crossable interfaces

It is assumed that all crossable interfaces are polygonal (i.e. they correspond to a surface).

In the sections that follow, $\mathcal{I}_c^{inc.}$ and $\mathcal{I}_c^{out.}$ refer respectively to the incoming and outgoing crossable interfaces into/from a traversable cell $c \in \mathcal{C}$. Since crossable interfaces are polygonal, the normal of interface $i \in \mathcal{I}$ oriented towards its destination traversable cell c_i^+ is referred to as $\overrightarrow{n_i}$.

Finally, the routing space can be represented by graph $G(\mathcal{C}, \mathcal{I})$ such that nodes are the traversable cells and edges are the existing crossable interfaces between these cells, as shown on Figure 8.4.

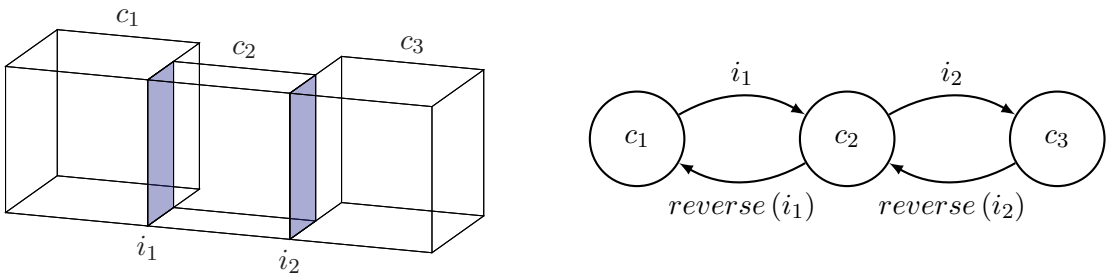


Figure 8.4 – A routing space $G(\mathcal{C}, \mathcal{I})$.

This graph is supposed to contain a path between each pair of cells in $\mathcal{C}$. If not, each connected component of the routing space must be considered separately. However, in this case, a solution to an instance of the CWRP exists only if the origin and destination cells are in the same connected component of $G(\mathcal{C}, \mathcal{I})$.

Assumption 3: Strong connectivity of the routing space

It is assumed that the routing space $G(\mathcal{C}, \mathcal{I})$ is strongly connected.

8.1.2 Construction of the traversable cells

In practice, a set of traversable cells $\mathcal{C}$ that avoids obstacles is not directly available. A methodology is proposed here to compute these cells from a satellite panel and the components fixed on it. Generally, a component is represented by a bounded polyhedron, while a wall is defined by a 3D-polygonal profile $\mathcal{S}_w$ and a normal $\overrightarrow{n_w}$. The methodology includes 4 steps detailed below.

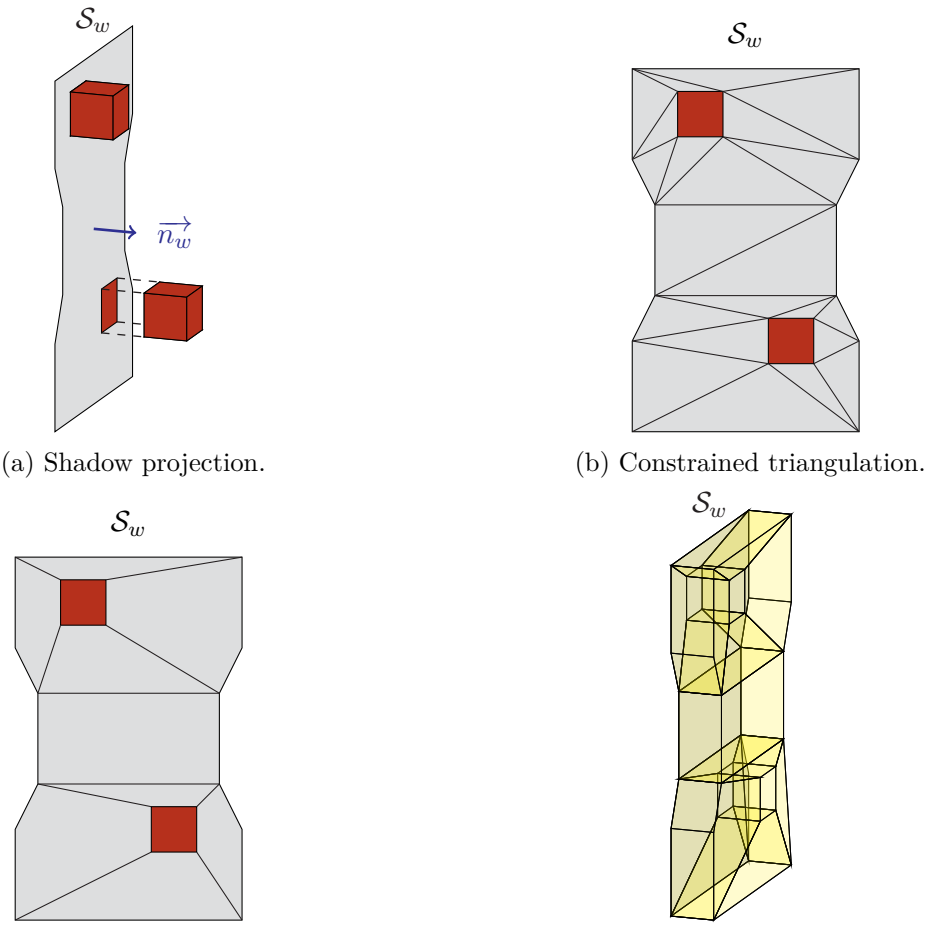
1. **Projection on the panel:** First, for each component, all vertices of the polyhedron are orthogonally projected on the plane that contains the panel surface. The convex hull of the projected points, which can be computed using KIRKPATRICK-SEIDEL's algorithm [56] or CHAN's algorithm [16], forms the *shadow* of the component on the panel (see Figure 8.5a).

2. **Constrained triangulation around shadows:** Shadows are inflated with a safety distance depending on the gauge of the waveguide. The goal is to ensure that the waveguide volume will never intersect a component by routing its neutral fibre in the adjacent

traversable cells. Then, the profile $\mathcal{S}_w$ of the wall is triangulated with the inflated shadows as constrained holes like in the work of [17], as illustrated on Figure 8.5b. Note that it may be necessary to use boolean operations (intersection, union, exclusion, disjunction) to merge overlapping or on-edge shadows.

3. **Triangle merge:** In order to reduce the complexity of the routing space, triangles resulting from the triangulation are merged into convex polygons (see Figure 8.5c). Remind that cells must be convex to allow using Linear Programming.

4. **Extrusion of traversable cells:** Last, the resulting convex polygons are inflated using the thickness of the routing space, providing the traversable cells of $\mathcal{C}$, as shown on Figure 8.5d. Again, the inflation must ensure that the waveguide volume will stay into the desired area. Moreover, it is also possible to create traversable cells below and above each component.



(a) Shadow projection.

(b) Constrained triangulation.

(c) Triangle merge.

(d) Cell extrusion.

Figure 8.5 – Construction of traversable cells $\mathcal{C}$.

Nevertheless, in an industrial context, there are many units on each panel of the satellite, which produces a very high number of traversable cells. So, in order to reduce the size and complexity of the routing space $G(\mathcal{C}, \mathcal{I})$, an iterative approach is used to route a waveguide. First, the corresponding FWRP instance is solved. If the resulting waveguide is out of the routing space or if it collides with components, routing space $G(\mathcal{C}, \mathcal{I})$ is computed with the previously

introduced method by considering only the overlapped components. Then, the CWRP is solved, generating a waveguide that avoids only the considered units. If it collides again with other components, a new routing space $G(\mathcal{C}, \mathcal{I})$ is built by adding these new conflicting units to the cell decomposition and another CWRP is solved, and so on until there is no more collision.

8.2 Definition of the CWRP

The Constrained Waveguide Routing Problem (CWRP) consists in routing in a detailed manner a single waveguide within a constrained three-dimensional space defined by a set of traversable cells. It is clearly an extension of the FWRP with space constraints. So, the waveguide π must satisfy the same constraints presented in Section 4.2.1 but with additional constraints which are specific to the CWRP and are introduced in Section 8.2.1 and Section 8.2.2. Then, the quality of a waveguide π is evaluated the same way as in the FWRP (see Section 4.2.2).

8.2.1 Routing space

The traversable cells of routing space $G(\mathcal{C}, \mathcal{I})$ describe the physical space that the neutral fibre $\mathcal{F}_\pi$ of waveguide π can cross. This can be translated by constraints on the waveguide segments. Indeed, each segment must be contained in the union of traversable cells, which can be written as follows:

$$[P_{\pi,k}, P_{\pi,k+1}] \subset \bigcup_{c \in \mathcal{C}} \mathcal{P}_c \quad \forall k \in [\![1, N_\pi]\!] \tag{8.1}$$

8.2.2 Wall-dependent attachability

As explained in Section 1.2.1, each segment of waveguide π must be fixed on the panel or wall it is routed on. The k^{th} segment is said to be *routed on* a panel if its neutral fibre $[P_{\pi,k}, P_{\pi,k+1}]$ intersects at least one of the traversable cells which have been generated on the panel. In order to ensure that the k^{th} segment can be fixed on the corresponding panel when it crosses a cell $c \in \mathcal{C}$, the orientation $o_{\pi,k}$ of the segment must be locally attachable. These constraints are called *wall-dependent attachability* and are said to be extrinsic because they depend on the walls on which the waveguides are routed.

Definition 26: Wall-dependent attachability

Let c be a traversable cell $\mathcal{C}$. The set of *locally attachable orientations* for cell c is referred to as $\mathcal{O}_c^{ext.} \subset \mathcal{O}$ and an orientation $o \in \mathcal{O}$ is said to be *locally attachable* in cell c if:

$$o \in \mathcal{O}_c^{ext.} \tag{8.2}$$

So, for waveguide π, the *wall-dependant attachability constraints* can be translated as:

$$[P_{\pi,k}, P_{\pi,k+1}] \cap \mathcal{P}_c \neq \varnothing \quad \Rightarrow \quad o_{\pi,k} \in \mathcal{O}_c^{ext.} \quad \forall k \in [\![1, N_\pi]\!] \tag{8.3}$$

In practice, orientation $o_{\pi,k}$ must ensure that at least one edge of the cross-section is orthogonal to the normal of the wall that corresponds to traversable cell $c \in \mathcal{C}$, as illustrated on Figure 8.6. So, in our case, the locally attachable orientations of cell c generated on a wall with normal $\overrightarrow{n_w}$ are defined by:

$$\mathcal{O}_c^{ext.} = \{o \in \mathcal{O} \mid (\overrightarrow{e_{o,x}} \cdot \overrightarrow{n_w} = 0) \vee (\overrightarrow{e_{o,y}} \cdot \overrightarrow{n_w} = 0)\} \tag{8.4}$$

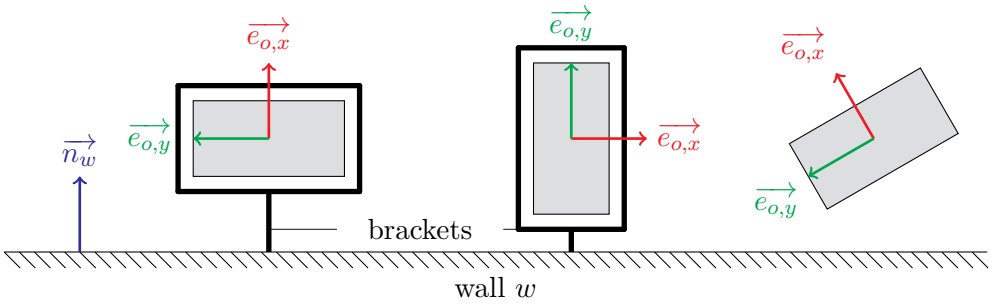


Figure 8.6 – From left to right, two attachable orientations and a non attachable one.

8.2.3 Connectivity

Last, the connectivity constraints for the origin and destination polyhedrons and orientations, introduced in Section 4.2.1, are extended. Indeed, waveguide π is assumed to start from a traversable cell $c^{ori.} \in \mathcal{C}$ and must reach a destination cell in the set $\mathcal{C}^{dest.} \subset \mathcal{C}$.

Assumption 4:

It is assumed that the origin and destination orientations are locally attachable:

$$o^{ori.} \in \mathcal{O}^{ext.}_{c^{ori.}} \tag{8.5}$$

$$\forall o \in \mathcal{O}^{dest.} \quad \forall c \in \mathcal{C}^{dest.} \qquad o \in \mathcal{O}^{ext.}_{c} \tag{8.6}$$

8.2.4 Full CWRP model

To sum up, the CWRP is an optimisation problem that can be written as follows:

$$\text{minimise } \gamma_\pi = \sum_{k=1}^{N_\pi - 1} \gamma_{b_{\pi,k}} + \mu \sum_{k=1}^{N_\pi} \ell_{\pi,k}$$

subject to:

$P_{\pi,1} \in \mathcal{P}^{ori.}$	Connectivity (position at origin)
$P_{\pi,N_\pi+1} \in \mathcal{P}^{dest.}$	Connectivity (position at destination)
$o_{\pi,1} = o^{ori.}$	Connectivity (orientation at origin)
$o_{\pi,N_\pi} \in \mathcal{O}^{dest.}$	Connectivity (orientation at destination)
$\pi \in \Pi(B_{cat.})$	Bend catalogue
$N_\pi \leq N_S$	Maximum number of bends
$L^{min} \leq L_{u_{\pi,k}} \quad \forall k \in [\![1, N_\pi]\!]$	Minimal length of straight sections
$o_{\pi,k} \in \mathcal{O}^{int.} \quad \forall k \in [\![1, N_\pi]\!]$	Global attachability
$[P_{\pi,k}, P_{\pi,k+1}] \subset \bigcup_{c \in \mathcal{C}} \mathcal{P}_c \quad \forall k \in [\![1, N_\pi]\!]$	Routing space
$[P_{\pi,k}, P_{\pi,k+1}] \cap \mathcal{P}_c \neq \varnothing \quad \Rightarrow \quad o_{\pi,k} \in \mathcal{O}^{ext.}_c \quad \forall k \in [\![1, N_\pi]\!]$	Wall-dependent attachability

Chapter 9

Resolution of the CWRP using Mixed Integer Linear Programming

A first way to solve the CWRP is to extend the Mixed Integer Linear Programming (MILP) formulation introduced in Chapter 5 for the resolution of the FWRP. To do so, the interfaces to cross inside the routing space are added as new decision steps. This modification requires to enumerate the candidate crossable interfaces but also has an impact on the possible orientation changes at each step, as explained in Section 9.1. Then, the proposed MILP model for the CWRP is presented in Section 9.2, while its experimentation on test instances is detailed in Section 9.3.

9.1 Input preprocessing

The previously studied FWRP consisted in building a sequence of at most $N_S - 1$ bends that makes it possible to reach the destination from the origin. By contrast, a solution to the CWRP defines also a path in the routing space $G(\mathcal{C}, \mathcal{I})$, called *routing channel*, in order to constrain the neutral fibre $\mathcal{F}_\pi$ to be inside the available space. A solution is then a sequence of decisions mixing the addition of bends from catalogue $B_{cat.}$ and the crossing of interfaces in the routing space. For instance, the sequence of decisions that consists in applying bend b_1 and then in successively crossing interfaces i_2 and i_2 allows reaching cell c_3 in the routing space presented on Figure 8.4 on page 107. Therefore, more than $N_S - 1$ decision steps are required and, at each step, either a bend can be applied or an interface crossed. Moreover, the space of candidate orientations must be modified by using the neutral bend $b_{neut.}$ even if the current orientation is not a destination one. To limit the size of the model, it is useful to precompute the possible current cells which can be obtained after k decision steps. Note that, in what follows, the maximum number of decision steps is fixed and referred to as K and satisfies $K \geq N_S - 1$.

9.1.1 Candidate orientations

In order to allow crossing an interface without applying a bend at a particular decision step $k \in [\![1, K]\!]$, the neutral bend $b_{neut.}$ must be a k-candidate orientation change, even if the current orientation o is not in $\mathcal{O}^{dest.}$. However, like for the FWRP, applying a neutral bend may lead to a solution only if a destination orientation in $\mathcal{O}^{dest.}$ can be reached using at most $K - k$ bends from catalogue $B_{cat.}$. To do so, Algorithm 9 on page 56 introduced for the FWRP is slightly modified into Algorithm 11. The latter uses the maximum number of decision steps K instead of the maximum number of segments N_S. The main update, on line 10, consists in adding $b_{neut.}$ as a k-candidate orientation change when $minBends(o) \leq K - k + 1$ rather than $o \in \mathcal{O}^{dest.}$.

Algorithm 11: Generate the constrained candidate orientation space $G\left(\mathcal{O}_1^{K+1}, \mathcal{R}_1^K\right)$

Input:

- Origin orientation: $o^{ori.}$
- Maximum number of decisions: K
- Kernel of reachable orientations: $G\left(\mathcal{O}_\infty, \mathcal{R}_\infty\right)$
- Distance to destination orientation: $minBends\left(o\right) \qquad \forall o \in \mathcal{O}_\infty$

$\mathcal{O}_1 \leftarrow \{o^{ori.}\}$
for $k \in [\![1, K]\!]$ **do**
  $\mathcal{O}_{k+1} \leftarrow \varnothing$
  $\mathcal{R}_k \leftarrow \varnothing$
  for $o \in \mathcal{O}_k$ **do**
    for $r \in \mathcal{R}_\infty^{out.}\left(o\right)$ **do**
      if $minBends\left(o_r^+\right) \leq K - k + 1$ **then**
        Add node o_r^+ in $\mathcal{O}_{k+1}$
        Add edge r in $\mathcal{R}_k$
    if $minBends\left(o\right) \leq K - k + 1$ **then**
      Add node o in $\mathcal{O}_{k+1}$
      Add edge $(o, o, b_{neut.})$ in $\mathcal{R}_k$
return $G\left(\mathcal{O}_1^{K+1}, \mathcal{R}_1^K\right)$

9.1.2 Candidate traversable cells

By analogy with the candidate orientation space $G\left(\mathcal{O}_1^{K+1}, \mathcal{R}_1^K\right)$, it is also possible to build a candidate cell space $G\left(\mathcal{C}_1^{K+1}, \mathcal{I}_1^K\right)$. To this end, Algorithm 11 is applied while replacing:

- the origin orientation $o^{ori.}$ by the origin traversable cell $c^{ori.}$;
- the kernel of reachable orientations $G\left(\mathcal{O}_\infty, \mathcal{R}_\infty\right)$ by the routing space $G\left(\mathcal{C}, \mathcal{I}\right)$;
- the distance to a destination orientation $minBends\left(o\right)$, from an orientation $o \in \mathcal{O}_\infty$, by the distance to a destination traversable cell $minDist\left(c\right)$, from a cell $c \in \mathcal{C}$.

This distance can be computed with Algorithm 8 by replacing again the kernel of reachable orientations $G\left(\mathcal{O}_\infty, \mathcal{R}_\infty\right)$ by the routing space $G\left(\mathcal{C}, \mathcal{I}\right)$ and using a cost function defined by $\gamma\left(i\right) = 1$ for interface $i \in \mathcal{I}$.

Furthermore, the construction of the candidate cell space $G\left(\mathcal{C}_1^{K+1}, \mathcal{I}_1^K\right)$ requires a crossable interface with a behaviour similar to the neutral bend $b_{neut.}$ in order to allow the addition of a bend without crossing an interface. This virtual crossable interface, called the *neutral interface*, can be defined as follows.

Definition 27: Neutral crossable interface

Let c be a traversable cell in $\mathcal{C}$. A *neutral interface* is a crossable interface $i^c_{neut.}$ which does not change the traversable cell. In other words, $i^c_{neut.}$ satisfies:

$$\mathcal{P}_{i^c_{neut.}} = \mathcal{P}_c \qquad c^-_{i^c_{neut.}} = c \qquad c^+_{i^c_{neut.}} = c$$

Note that the polyhedron of a neutral interface is not a surface.

9.1.3 Maximum number of decision steps K

It is difficult to predict the number of decision steps required to find an optimal waveguide inside the routing space. However, it is known that at least $minBends\left(o^{ori.}\right)$ bends must be added and $minDist\left(c^{ori.}\right)$ interfaces must be crossed to solve the CWRP. A lower bound of the necessary number of decision steps is then $minBends\left(o^{ori.}\right) + minDist\left(c^{ori.}\right)$. In practice, the optimum is looked for among solutions that use at most $N_S - 1$ bends, so the maximum number of decision steps K is generally chosen greater than $N_S + minDist\left(c^{ori.}\right) - 1$.

9.2 MILP formulation

The CWRP can be formulated as a MILP problem which consists in choosing the bends of waveguide π and the interfaces crossed by each of its segments as well as in defining the lengths of the straight sections between these bends. In what follows, it is assumed that the routing space $G\left(\mathcal{C}, \mathcal{I}\right)$ is not empty.

9.2.1 Trivial case $N_C = 1$

When there is an only one traversable cell c in the routing space, that means $N_C = 1$, the CWRP can be solved with the same MILP model as for the FWRP by adding the following space constraints on the vertices of neutral fibre $\mathcal{F}_\pi$:

$$p_k \in \mathcal{P}_c \quad \forall k \in [\![2, N_S]\!] \tag{9.1}$$

The first and last points are not constrained to be respectively in the origin and destination traversable cells $c^{ori.}$ and $\mathcal{C}^{dest.}$ because there are already constrained by the origin and destination polyhedrons $\mathcal{P}^{ori.}$ and $\mathcal{P}^{dest.}$.

9.2.2 General case with $N_C \geq 2$

In the general case with $N_C \geq 2$, the MILP model of the FWRP is not sufficient and must be extended with space constraints. Indeed, discrete variables must be introduced to choose the interfaces crossed by each segment of waveguide π. More precisely, the CWRP can be formulated as a MILP model that contains five kinds of variables:

- *bend variables* $x_{k,r}$ such that, for $k \in [\![1, K]\!]$ and $r \in \mathcal{R}_k$, integer variable $x_{k,r}$ takes value 1 if orientation change r is applied at the end point of the k^{th} segment of neutral fibre $\mathcal{F}_\pi$ (that means the k^{th} and $k+1^{\text{th}}$ segments have respectively an orientation o^-_r and o^+_r), 0 otherwise;

- *interface variables* $y_{k,i}$ such that, for $k \in [\![1, K]\!]$ and $i \in \mathcal{I}_k$, integer variable $y_{k,i}$ takes value 1 if interface i is crossed by the k^{th} segment of neutral fibre $\mathcal{F}_\pi$ (that means the $k+1^{\text{th}}$ point of neutral fibre $\mathcal{F}_\pi$ is on crossable interface i), 0 otherwise;

- *straight section variables* $z_{k,k'}$ such that, for $k \in [\![1, K+1]\!]$ and $k' \in [\![k+1, K+2]\!]$, integer variable $z_{k,k'}$ takes value 1 if there is a straight section between the k^{th} and k'^{th} points of neutral fibre $\mathcal{F}_\pi$, that is to say non-neutral bends are applied at these points and neutral bends are applied between them, 0 otherwise;

- *length variables* ℓ_k such that, for $k \in [\![1, K+1]\!]$, real variable ℓ_k is the length of the k^{th} segment of neutral fibre $\mathcal{F}_\pi$;

- *position variables* $p_k = (p_{k,x}, p_{k,y}, p_{k,z})$ such that, for $k \in [\![1, K+2]\!]$, real variable $p_{k,x}$ (respectively $p_{k,y}$ and $p_{k,z}$) is the x-coordinate (respectively y-coordinate and z-coordinate) of the k^{th} point of neutral fibre $\mathcal{F}_\pi$.

With these decision variables, a MILP formulation of the CWRP can be written as follows:

$$\text{minimise} \sum_{k=1}^{K} \sum_{r \in \mathcal{R}_k} \gamma_{b_r} x_{k,r} + \mu \sum_{k=1}^{K+1} \ell_k \tag{9.2}$$

subject to:

$$\sum_{r \in \mathcal{R}_k} x_{k,r} = 1 \qquad \forall k \in [\![1, K]\!] \tag{9.3}$$

$$\sum_{r \in \mathcal{R}_k^{out.}(o)} x_{k+1,r} = \sum_{r \in \mathcal{R}_k^{inc.}(o)} x_{k,r} \qquad \forall k \in [\![1, K-1]\!],\ \forall o \in \mathcal{O}_k \tag{9.4}$$

$$\sum_{k=1}^{K} \sum_{\substack{r \in \mathcal{R}_k \\ b_r \neq b_{neut.}}} x_{k,r} \leq N_S - 1 \tag{9.5}$$

$$\sum_{i \in \mathcal{I}_k} y_{k,i} = 1 \qquad \forall k \in [\![1, K]\!] \tag{9.6}$$

$$\sum_{i \in \mathcal{I}_k^{out.}(c)} y_{k+1,i} = \sum_{i \in \mathcal{I}_k^{inc.}(c)} y_{k,i} \qquad \forall k \in [\![1, K-1]\!],\ \forall c \in \mathcal{C}_k \tag{9.7}$$

$$y_{k,i} + y_{k+1,reverse(i)} \leq 1 \qquad \forall k \in [\![1, K-1]\!],\ \forall i \in \mathcal{I}_k \setminus i_{neut.} \tag{9.8}$$

$$p_1 \in \mathcal{P}^{ori.} \tag{9.9}$$

$$p_{K+2} \in \mathcal{P}^{dest.} \tag{9.10}$$

$$\overrightarrow{p_1 p_2} = \ell_1 \overrightarrow{e_{o^{ori.},z}} \tag{9.11}$$

$$a_q p_{k+1,x} + b_q p_{k+1,y} + c_q p_{k+1,z} + d_q \leq M_q (1 - y_{k,i}) \qquad \forall k \in [\![1, K]\!],\ \forall i \in \mathcal{I}_k,\ \forall q \in \mathcal{Q}_{\mathcal{P}_i} \tag{9.12}$$

$$\sum_{k'=k+1}^{K+2} z_{k,k'} + \sum_{\substack{r \in \mathcal{R}_{k-1} \\ b_r = b_{neut.}}} x_{k-1,r} = 1 \qquad \forall k \in [\![1, K+1]\!] \tag{9.13}$$

$$\sum_{k'=1}^{k-1} z_{k',k} + \sum_{\substack{r \in \mathcal{R}_{k-1} \\ b_r = b_{neut.}}} x_{k-1,r} = 1 \qquad \forall k \in [\![2, K+2]\!] \tag{9.14}$$

$$\sum_{k''=k}^{k'-1} \ell_{k''} \geq L^{min} + \sum_{r \in \mathcal{R}_{k-1}} L_{b_r} x_{k-1,r} \qquad \forall k \in [\![1, K+1]\!],\ \forall k' \in [\![k+1, K+2]\!]$$
$$+ \sum_{r \in \mathcal{R}_{k'-1}} L_{b_r} x_{k'-1,r} - M_{len.} (1 - z_{k,k'}) \tag{9.15}$$

$$\overrightarrow{p_k p_{k+1}} \leq \ell_k \overrightarrow{e_{o_r^+,z}} + (1 - x_{k-1,r}) \overrightarrow{M_{succ.}} \qquad \forall k \in [\![2, K+1]\!],\ \forall r \in \mathcal{R}_{k-1} \tag{9.16}$$

$$\overrightarrow{p_k p_{k+1}} \geq \ell_k \overrightarrow{e_{o_r^+,z}} - (1 - x_{k-1,r}) \overrightarrow{M_{succ.}} \qquad \forall k \in [\![2, K+1]\!],\ \forall r \in \mathcal{R}_{k-1} \tag{9.17}$$

$$\sum_{\substack{r \in \mathcal{R}_k \\ b_r = b_{neut.}}} x_{k,r} + \sum_{\substack{i \in \mathcal{I}_k \\ i = i_{neut.}}} y_{k,i} \geq 1 \qquad \forall k \in [\![1, K]\!] \tag{9.18}$$

$$\sum_{\substack{r \in \mathcal{R}_k \\ b_r = b_{neut.}}} x_{k,r} + \sum_{\substack{i \in \mathcal{I}_k \\ i = i_{neut.}}} y_{k,i} \leq \sum_{\substack{r \in \mathcal{R}_{k+1} \\ b_r = b_{neut.}}} x_{k+1,r} + \sum_{\substack{i \in \mathcal{I}_{k+1} \\ i = i_{neut.}}} y_{k+1,i} \qquad \forall k \in [\![1, K-1]\!] \tag{9.19}$$

$$x_{k,r} \in \{0,1\} \quad \forall k \in [\![1,K]\!],\ \forall r \in \mathcal{R}_k \tag{9.20}$$
$$y_{k,i} \in \{0,1\} \quad \forall k \in [\![1,K]\!],\ \forall i \in \mathcal{I}_k \tag{9.21}$$
$$z_{k,k'} \quad \forall k \in [\![1,K+1]\!],\ k' \in [\![k+1,K+2]\!] \tag{9.22}$$
$$\ell_k \in \mathbb{R}^+ \quad \forall k \in [\![1,K+1]\!] \tag{9.23}$$
$$p_k \in \mathbb{R}^3 \quad \forall k \in [\![1,K+2]\!] \tag{9.24}$$

Most constraints as well as the criterion are the same as in the MILP model proposed for the FWRP (see Section 5.2). The specific constraints of the CWRP are detailed in the following sections. In particular, by contrast with the FWRP, a valid formulation of the minimal length for straight sections requires the introduction of the straight sections variables. Three parts can be identified in this MILP formulation:

- an *orientation sub-problem* (with Constraints 9.3-9.5) which corresponds to a multiple target shortest path problem in the space of candidate orientations $G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$;
- a *channel sub-problem* (with Constraints 9.6-9.8) which corresponds to a multiple target shortest path problem in the routing space $G(\mathcal{C}, \mathcal{I})$;
- a *3D-position sub-problem* (with Constraints 9.9-9.19) which defines the restrictions on the vertices and lengths of the neutral fibre.

The three sub-problems are coupled through a set of *coupling constraints* composed of Constraints 9.12-9.19 that link position, length, bend, interface and straight section variables.

It can be noted that the proposed model can use more than N_S segments for the neutral fibre $\mathcal{F}_\pi$ since $K \geq N_S - 1$. However, remind that, at each decision step, either a bend can be applied or an interface may be crossed, which is ensured by Constraints 9.18. If the addition of a bend creates a new waveguide segment, on the opposite, the crossing of an interface does not create one since the orientation is not modified. It results that each time an interface is crossed, the following segment is only an extension of the previous one. Finally, the number of segments is limited to N_S thanks to Constraints 9.5.

Note that the waveguide π can be rebuilt from a solution of the MILP model in the same way as for the FWRP, using Algorithm 10 on page 59.

9.2.3 Bend sequence sub-problem

Like for the FWRP, the *bend sequence sub-problem* aims at finding a bend combination that allows waveguide π to reach a destination orientation in $\mathcal{O}^{dest.}$ from the origin orientation $o^{ori.}$. It can be written as follows:

$$\begin{aligned}
&\text{minimise} \sum_{k=1}^{K} \sum_{r \in \mathcal{R}_k} \gamma_{b_r} x_{k,r} \\
&\text{subject to:} \\
&\sum_{r \in \mathcal{R}_k} x_{k,r} = 1 && \forall k \in [\![1,K]\!] \\
&\sum_{r \in \mathcal{R}_k^{out.}(o)} x_{k+1,r} = \sum_{r \in \mathcal{R}_k^{inc.}(o)} x_{k,r} && \forall k \in [\![1,K-1]\!],\ \forall o \in \mathcal{O}_k \\
&\sum_{k=1}^{K} \sum_{\substack{r \in \mathcal{R}_k \\ b_r \neq b_{neut.}}} x_{k,r} \leq N_S - 1 \\
&x_{k,r} \in \{0,1\} && \forall k \in [\![1,K]\!],\ \forall r \in \mathcal{R}_k
\end{aligned}$$

It is pretty similar to the bend sequence sub-problem of the FWRP, except that the symmetry Constraints 5.11 have been replaced by Constraint 9.5 on the maximum number of non-neutral bends. Indeed, in the CWRP, a neutral bend can be used at each decision step and not only at the end of the bend sequence. Apart from that, Constraints 9.3 and 9.4 are equivalent to Constraints 5.9 and 5.10 which have been presented in Section 5.2.3 on page 59.

9.2.4 Channel sub-problem

Definition

The *channel sub-problem* aims at building a path that allows reaching a destination cell in $\mathcal{C}^{dest.}$ from the origin cell $c^{ori.}$. It is a multiple target shortest path problem in the graph of orientations $G\left(\mathcal{C}_1^{K+1}, \mathcal{I}_1^K\right)$ and can be written as follows:

$$\begin{aligned}
\sum_{i \in \mathcal{I}_k} y_{k,i} &= 1 && \forall k \in [\![1, K]\!] \\
\sum_{i \in \mathcal{I}_k^{out.}(c)} y_{k+1,i} &= \sum_{i \in \mathcal{I}_k^{inc.}(c)} y_{k,i} && \forall k \in [\![1, K-1]\!],\ \forall c \in \mathcal{C}_k \\
y_{k,i} + y_{k+1,reverse(i)} &\leq 1 && \forall k \in [\![1, K-1]\!],\ \forall i \in \mathcal{I}_k \setminus i_{neut.} \\
y_{k,i} &\in \{0,1\} && \forall k \in [\![1, K]\!],\ \forall i \in \mathcal{I}_k
\end{aligned} \tag{9.25}$$

It is similar to the bend sequence sub-problem but adapted to the routing space. Constraints 9.6 ensure that exactly one interface is crossed by the k^{th} segment of the neutral fibre $\mathcal{F}_\pi$. As the last candidate traversable cell set $\mathcal{C}_{K+1}$ contains only destination cells, they also ensure that a destination cell is reached. Then, like in the natural linear programming formulation of the shortest path problem, Constraints 9.7 enforce each k-candidate cell $c \in \mathcal{C}_k$ to have as many outgoing crossed interfaces as incoming ones, except for the origin and destination cells.

Symmetry breaking for directly revisiting traversable cells

$$y_{k,i} + y_{k+1,reverse(i)} \leq 1 \quad \forall k \in [\![1, K-1]\!],\ \forall i \in \mathcal{I}_k \mid i \neq i_{neut.}$$

If the routing space contains cramped cells, it may be necessary to visit a traversable cell more than once to solve the CWRP. Indeed, it is often difficult to place several bends inside a narrow cell. In this case, a detour by a larger adjacent cell can help reaching the wanted orientation inside the cramped cell. However, directly revisiting a traversable cell $c \in \mathcal{C}$ without having applied any orientation change or visited another cell is useless and only generates symmetry in the space of solutions. Constraints 9.8 allow to break this symmetry by forbidding the direct revisit of a traversable cell without having applied an orientation change or visited another cell.

9.2.5 3D-position sub-problem and coupling constraints

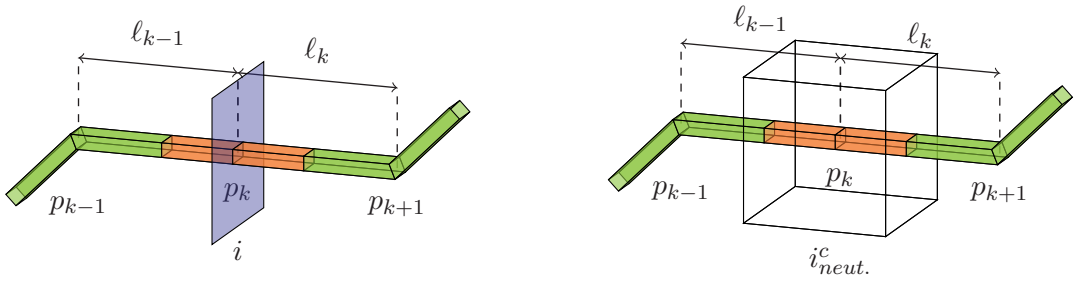
Constraints 9.9, 9.10, 9.11, 9.16 and 9.17 are respectively equivalent to Constraints 5.12, 5.13, 5.14, 5.16 and 5.17 which have been presented in Section 5.2.4 on page 60. The following sections introduce Constraints 9.12-9.15 and Constraints 9.18-9.19 that are new in comparison with the FWRP.

Interface constraints

$$a_q p_{k+1,x} + b_q p_{k+1,y} + c_q p_{k+1,z} + d_q \leq M_q \left(1 - y_{k+1,i}\right) \quad \forall k \in [\![1, K]\!],\ \forall i \in \mathcal{I}_k,\ \forall q \in \mathcal{Q}_{\mathcal{P}_i}$$

Constraints 9.12 use the big-M technique on the interfaces variables. These constraints force the k^{th} point of neutral fibre $\mathcal{F}_\pi$ to be located inside polyhedron $\mathcal{P}_i$ if interface $i \in \mathcal{I}_k$ is crossed

($y_{k+1,i} = 1$), as shown in Figure 9.1. Note that when the neutral interface $i^c_{neut.}$ is crossed, the k^{th} point is inside the polyhedron $\mathcal{P}_c$ of the associated traversable cell $c \in \mathcal{C}$. On the opposite if $y_{k+1,i} = 0$, these constraints must be disabled by choosing big-M parameter M_q in such a way that the position of point p_{k+1} is not limited. A valid M_q-value is proposed in Property 10.



(a) Non neutral interface.

(b) Neutral interface.

Figure 9.1 – Portion of a pipe crossing an interface (the straight section with variable length is depicted in orange).

Property 10: Big-M value in interface constraints

For $k \in [\![1, K]\!]$, $i \in \mathcal{I}_k$ and $q \in \mathcal{Q}_{\mathcal{P}_i}$, Constraints 9.12 are disabled when $y_{k+1,i} = 0$ in the sense that they do not constrain the position of point p_{k+1} if:

$$M_q = \max_{p \in \mathcal{P}_i} (a_q p_x + b_q p_y + c_q p_z + d_q)$$

The proposed M_q-values can be easily precomputed using the minimal and maximal values of coordinates p_x, p_y and p_z for a point p in each polyhedron $\mathcal{P}_i$.

Straight section constraints

$$\sum_{k'=k+1}^{K+2} z_{k,k'} + \sum_{\substack{r \in \mathcal{R}_{k-1} \\ b_r = b_{neut.}}} x_{k-1,r} = 1 \quad \forall k \in [\![1, K+1]\!]$$

$$\sum_{k'=1}^{k-1} z_{k',k} + \sum_{\substack{r \in \mathcal{R}_{k-1} \\ b_r = b_{neut.}}} x_{k-1,r} = 1 \quad \forall k \in [\![2, K+2]\!]$$

Constraints 9.13 and 9.14 ensure that at each decision step either a neutral bend is applied or exactly one straight section is started and another one is ended, which corresponds to the application of a non-neutral bend. For the borderline case $k = 1$, Constraints 9.13 guarantee that exactly one straight section is started from the first vertex of neutral fibre $\mathcal{F}_\pi$ since $\mathcal{R}_0 = \varnothing$. Similarly, for the borderline case $k = K+2$, Constraints 9.14 guarantee that exactly one straight section is ended at the last vertex of neutral fibre $\mathcal{F}_\pi$ since $\mathcal{R}_{N_S} = \varnothing$. Furthermore, when there is a straight section between the k^{th} and k'^{th} points of neutral fibre $\mathcal{F}_\pi$ (that is to say when $z_{k,k'} = 1$), these constraints force all intermediate bends to be neutral (a mathematical demonstration is proposed in the appendices).

Minimum length of straight sections

$$\sum_{k''=k}^{k'-1} \ell_{k''} \geq L^{min} + \sum_{r \in \mathcal{R}_{k-1}} L_{b_r} x_{k-1,r} \qquad \forall k \in [\![1, K+1]\!],\ \forall k' \in [\![k+1, K+2]\!]$$
$$+ \sum_{r \in \mathcal{R}_{k'-1}} L_{b_r} x_{k'-1,r} - M_{len.}\,(1 - z_{k,k'})$$

Constraints 9.15 impose a minimal length on straight sections. In the FWRP, the expression of these constraints (see Constraints 5.15) is simple because all neutral bends are applied at the end of the decision sequence and there is no space constraint on the vertices of neutral fibre $\mathcal{F}_\pi$, except succession constraints. It results that the minimal length of straight sections can be expressed as a minimal length on each segment by adapting the length contribution $L_{b_{neut.}}$ for neutral bends (see Section 5.2.4). However, in the CWRP, these assumptions do not hold since it is possible to alternate neutral and non-neutral bends, and the vertices are constrained to be on interfaces. The expression of the minimal length on straight sections should consider the total length of segments between two non-neutral bends, and not only the length of each segment. To do so, Constraints 5.15 of the FWRP are reformulated into Constraints 9.15 using the big-M technique on the straight section variables that define the position of the straight sections. These constraints force the sum of segment lengths between the k^{th} and k'^{th} points of neutral fibre $\mathcal{F}_\pi$ to satisfy the minimal length of straight sections if $z_{k,k'} = 1$. On the opposite if $z_{k,k'} = 0$, these constraints must be disabled by choosing big-M parameter $M_{len.}$ so that the total length of the segments between the k^{th} and k'^{th} points is not limited. An obvious valid $M_{len.}$-value is proposed in Property 11.

Property 11: Big-M value in minimal straight section length constraints

For $k \in [\![1, K+1]\!]$ and $k' \in [\![k+1, K+2]\!]$, Constraints 9.15 are disabled when $z_{k,k'} = 0$ in the sense that they do not constrain the total length of the segments between the k^{th} and k'^{th} points if:

$$M_{len.} = L^{min} + 2 \max_{b \in B_{cat.}} (L_b)$$

Decision unicity

$$\sum_{\substack{r \in \mathcal{R}_k \\ b_r = b_{neut.}}} x_{k,r} + \sum_{\substack{i \in \mathcal{I}_k \\ i = i_{neut.}}} y_{k,i} \geq 1 \quad \forall k \in [\![1, K]\!]$$

Constraints 9.18 ensure that a non-neutral bend cannot be applied at the same decision step when a non-neutral interface is crossed (and reciprocally). However, a neutral bend can be applied at the same time when a neutral interface is crossed.

Symmetry breaking for neutral bends and interfaces

$$\sum_{\substack{r \in \mathcal{R}_k \\ b_r = b_{neut.}}} x_{k,r} + \sum_{\substack{i \in \mathcal{I}_k \\ i = i_{neut.}}} y_{k,i} \leq \sum_{\substack{r \in \mathcal{R}_{k+1} \\ b_r = b_{neut.}}} x_{k+1,r} + \sum_{\substack{i \in \mathcal{I}_{k+1} \\ i = i_{neut.}}} y_{k+1,i} \quad \forall k \in [\![1, K-1]\!]$$

Like in Section 5.2.3, due to the introduction of the neutral bend $b_{neut.}$ and of the neutral interfaces $i_{neut.}$, it is possible to have neutral decision steps where both a neutral bend and a neutral interface are chosen. A solution decision sequence which contains a neutral decision

step can be translated into several sequences of decisions. For instance, if the decision sequence $[(b_1, i_{neut.}), (b_{neut.}, i_1), (b_{neut.}, i_{neut.})]$ is a solution, then $[(b_1, i_{neut.}), (b_{neut.}, i_{neut.}), (b_{neut.}, i_1)]$ and $[(b_{neut.}, i_{neut.}), (b_1, i_{neut.}), (b_{neut.}, i_1)]$ are solutions too and have the same cost. Constraints 9.19 break this symmetry by forcing the neutral decision steps to be applied at the end of the decision sequence. To do so, if a neutral bend and a neutral interface are chosen at step $k \in [\![1, K-1]\!]$, then a neutral bend and a neutral interface must be chosen at step $k+1$. By induction, it can be easily proved that neutral decisions must also be used at all the following steps of the decision sequence. In this case, only $[(b_1, i_{neut.}), (b_{neut.}, i_1), (b_{neut.}, i_{neut.})]$ is a valid sequence of decisions.

9.3 Experiments on the CWRP

In this Section, the MILP formulation of the CWRP is experimented on the simple test instances presented in Section 10.3.1. The number of variables and constraints of the corresponding MILP models are reported in Section 9.3.2. Last, the performances of this approach to solve the CWRP are studied in Section 9.3.3.

9.3.1 Instance sets

The MILP formulation of the CWRP has been experimented on three instance sets using previously introduced bend catalogues $B^{90°}_{cat.}$, $B^{45°}_{cat.}$ and $B^{30°}_{cat.}$ (see Section 5.3.1 on page 63). Remind that these catalogues contain respectively 5, 9 and 17 bends and are detailed in Table 11 in the appendices. Again, the minimal length is $L^{min} = 5$, the linear cost is unit, $\mu = 1$, and each bend $b \in B_{cat.}$ of the catalogue has a cost $\gamma_b = 100$, except for the twist which has a cost $\gamma_b = 1000$. For each bend catalogue $B_{cat.}$, the instance set contains four instances with gradual difficulty. All instances use a simple routing space with the traversable cells described in Table 7 on page 168. It is made of 8 cells with 16 interfaces and its dimensions are $1450 \times 550 \times 550$. Origin and destination configurations are detailed in Table 8 on page 168. All instances share the same origin configuration $\theta^{ori.}$ but the destination $\theta^{dest.}$ is different and moving further from the origin at each following instance making the problem harder to solve. The maximum number of segments is $N_S = 11$, which corresponds to a sufficient number of bends to make each instance feasible. An illustration of these instances is given by Figure 9.2.

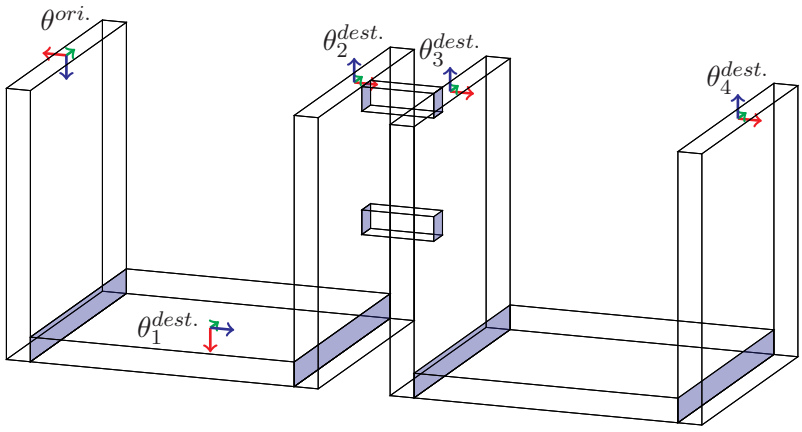


Figure 9.2 – Instances of the CWRP for the MILP approach.

These instances are extremely simple and not representative of the complexity of a satellite payload. However, they are used in order to validate the MILP formulation and to ensure that

they can be efficiently solved with this approach, especially the ones using the industrial bend catalogue $B_{cat.}^{30°}$. In practice, such instances must be solved within a few seconds in order to make an industrial use possible.

9.3.2 MILP model sizes

Figure 9.3 presents the MILP model sizes for the test instances in terms of number of variables and constraints. The maximum number of decision steps K used for each instance is also reported. By comparison with the FWRP MILP models (see Section 9.3.2), the required number of variables for the CWRP has grown by several orders of magnitude, which significantly increases the combinatorics. Obviously, this is due to the introduction of variables related to the interfaces to cross. However, another reason is that the maximum number of decisions steps K must be sufficiently large to allow crossing enough interfaces to reach the destination but also applying the bends needed to do so. As a consequence, many more orientation changes from the kernel $G(\mathcal{O}_\infty, \mathcal{R}_\infty)$ as well as interfaces from the routing space $G(\mathcal{C}, \mathcal{I})$ are candidates since K-length paths are considered (here $K = N_S + minDist(c^{ori.}) - 1$ is used). In comparison, the FWRP only considers $N_S - 1$ decision steps.

		Catalogue $B_{cat.}^{90°}$		Catalogue $B_{cat.}^{45°}$		Catalogue $B_{cat.}^{30°}$	
#	K	# variables	# constraints	# variables	# constraints	# variables	# constraints
1	11	1349	7820	5729	34698	51605	314708
2	12	1548	8992	6520	39494	60348	367912
3	14	1921	11145	8333	50463	84817	516801
4	16	2264	13066	10116	61200	109256	665458

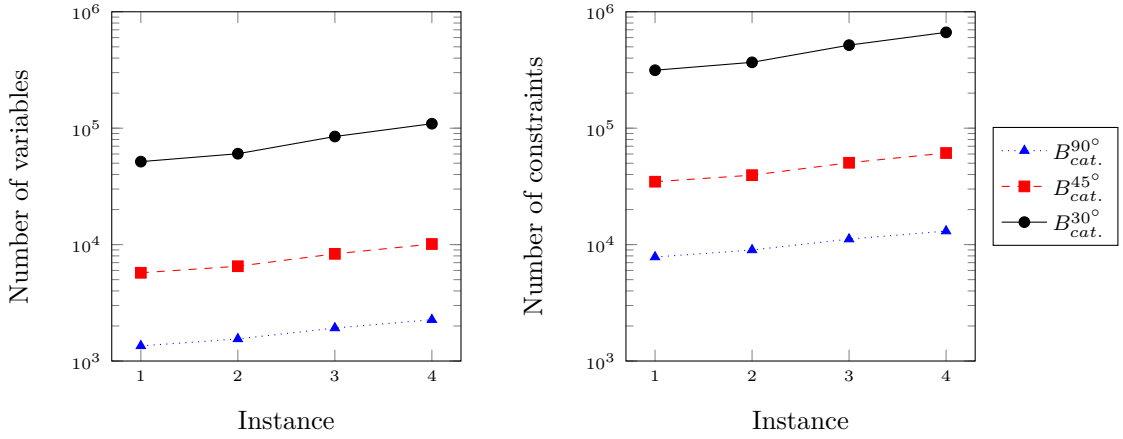


Figure 9.3 – Size of the CWRP MILP models.

9.3.3 Test instances resolution

The experiments exposed here have also been performed using the configuration presented in Section 5.3.4 on page 65. The resolution has been stopped after 1 hour. Figure 9.4 shows the evolution of the percentage of instances solved with regards to the runtime, for each catalogue, and Table 9.1, Table 9.2 and Table 9.3 present the results obtained for catalogues $B_{cat.}^{90°}$, $B_{cat.}^{45°}$ and $B_{cat.}^{30°}$ respectively.

Like for the FWRP, a resolution stopped when reaching the optimality or the time limit and another one stopped when meeting the first solution are studied. As expected, the solutions found with the MILP formulation are acceptable from a designer's point of view, as illustrated on Figure 9.5.

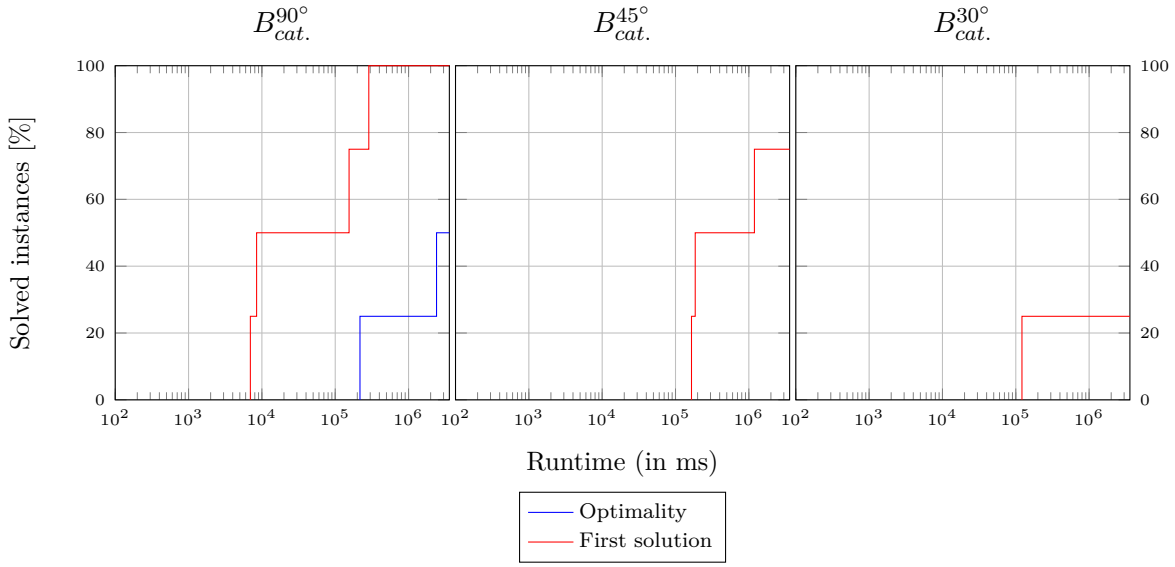


Figure 9.4 – Evolution of success rates with MILP for the CWRP.

Inst.		MILP optimal						MILP first					
#	Opti. cost	Cost	Gap [%]	Linear cost	Bend cost	Runtime (ms)	Iter.	Cost	Gap [%]	Linear cost	Bend cost	Runtime (ms)	Iter.
1	925	**925**	**0**	**825**	**100**	**217558**	**137616**	**925**	**0**	**825**	**100**	**6963**	**465**
2	1800	**1800**	**0**	**1600**	**200**	**2408208**	**1339183**	**1800**	**0**	**1600**	**200**	**8471**	**2225**
3	-	-	-	-	-	**3600093**	**2209691**	**3621.2**	-	**1921.2**	**1700**	**154501**	**40772**
4	-	-	-	-	-	**3600108**	**1976226**	**3660.6**	-	**2860.6**	**800**	**287050**	**103373**

Table 9.1 – Results for the CWRP using MILP on $B^{90°}_{cat.}$ instances.

Inst.		MILP optimal						MILP first					
#	Opti. cost	Cost	Gap [%]	Linear cost	Bend cost	Runtime (ms)	Iter.	Cost	Gap [%]	Linear cost	Bend cost	Runtime (ms)	Iter.
1	-	-	-	-	-	3600747	177426	**925**	**-**	**825**	**100**	**165218**	**829**
2	-	-	-	-	-	3600231	195229	**1800**	**-**	**1600**	**200**	**185713**	**2491**
3	-	-	-	-	-	3600367	122637	**2200**	**-**	**1800**	**400**	**1191281**	**16676**
4	-	-	-	-	-	3602939	75067	-	-	-	-	3600778	87136

Table 9.2 – Results for the CWRP using MILP on $B^{45°}_{cat.}$ instances.

Inst.		MILP optimal						MILP first					
#	Opti. cost	Cost	Gap [%]	Linear cost	Bend cost	Runtime (ms)	Iter.	Cost	Gap [%]	Linear cost	Bend cost	Runtime (ms)	Iter.
1	-	-	-	-	-	3653553	58033	**925**	**-**	**825**	**100**	**121373**	**3031**
2	-	-	-	-	-	3779440	29106	-	-	-	-	3710041	29035
3	-	-	-	-	-	3772696	34008	-	-	-	-	3608123	65358
4	-	-	-	-	-	3620041	23122	-	-	-	-	3603771	46052

Table 9.3 – Results for the CWRP using MILP on $B^{30°}_{cat.}$ instances.

However, only a few instances using the simplest catalogue $B^{90°}_{cat.}$ can be optimally solved within the time limit and reaching optimality requires an excessive runtime: even the second instance requires dozens of minutes to be solved. Moreover, although instances are very simple, no optimal solution can be found for the cases using the more realistic catalogues $B^{45°}_{cat.}$ and $B^{30°}_{cat.}$. Worse, even finding a first solution requires several minutes. As a consequence, the CWRP cannot be solved with the current MILP formulation and a more efficient approach must be proposed.

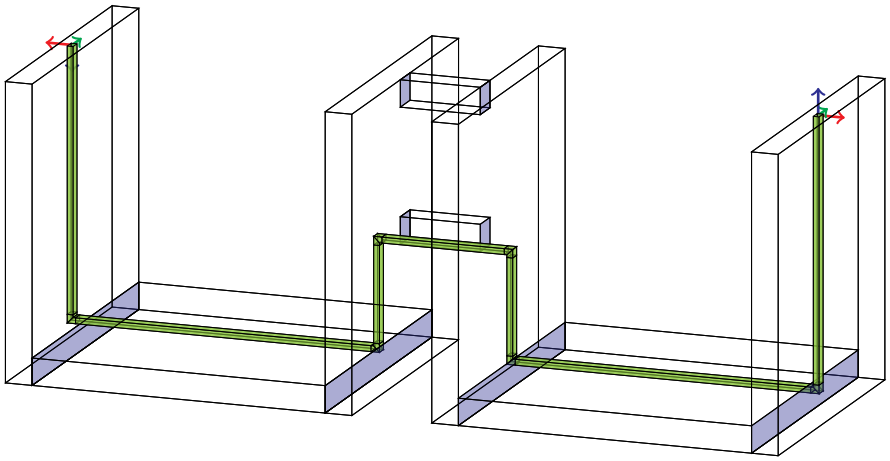


Figure 9.5 – Example of waveguide routed with MILP on instance 4.

Chapter 10

Resolution of the CWRP using Informed Search Algorithms

A formulation of the CWRP adapted to Search Algorithms (SAs) is proposed in this chapter. It is based on an extension of the notion of routing plan, introduced in Section 10.1, that describes a partially routed waveguide inside the routing space. The feasibility of a routing plan is then evaluated using Linear Programming (LP) while its successors are easily expressed through routing decisions like the addition of a bend or the crossing of an interface. Heuristic evaluation functions which take space constraints into account are proposed in Section 10.2, making the use of ISAs possible to solve the CWRP. Last, the Search Problem (SP) formulation is experimented in Section 10.3 on various test instances.

10.1 Routing plan formulation

As for the FWRP, a waveguide π can be routed in the routing space $G(\mathcal{C}, \mathcal{I})$ by building iteratively its neutral fibre $\mathcal{F}_\pi$. To do so, the neutral fibre is extended from the origin configuration $\theta^{ori.}$ inside the origin cell $c^{ori.}$ by making at each step decisions like the addition of a bend from catalogue $B_{cat.}$ or the crossing of an interface in routing space $G(\mathcal{C}, \mathcal{I})$. To formalise the approach, the concept of routing plan, introduced in Section 6.1 on page 71, is extended with the space decisions made so far on the waveguide components.

10.1.1 Routing plan

Definition 28: Routing plan (in constrained space)

In constrained space, a *routing plan* s describes, in an abstract way, a neutral fibre $\mathcal{F}_s$ composed of N_s successive segments with, for each segment k (see Figure 10.1):

- the orientation change $r_k^s \in \mathcal{R}_k$ applied at the end point of segment k of neutral fibre $\mathcal{F}_s$, for $k \in [\![1, N_s - 1]\!]$;
- the sequence of interfaces $\mathcal{I}_k^s \subseteq \mathcal{I}$ crossed by segment k of neutral fibre $\mathcal{F}_s$, for $k \in [\![1, N_s]\!]$.

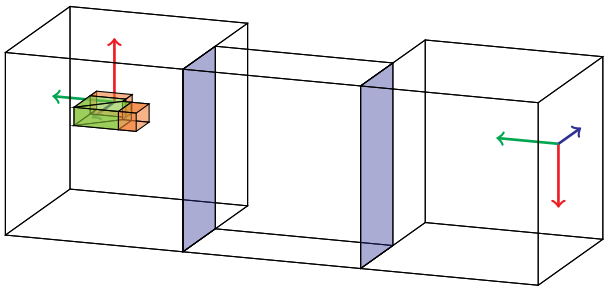
Moreover, a routing plan can be terminated or not, characterised by a boolean $Term^s$. When routing plan s is terminated, neutral fibre $\mathcal{F}_s$ has to reach the destination polyhedron $\mathcal{P}^{dest.}$.
The set of routing plans is denoted by $\mathcal{S}$.

Several data can be derived from the basic definition of a routing plan:

- the bend $b_k^s \in B_{cat.}$ applied at the end point of the k^{th} segment can be deduced from the sequence of orientation changes (see Section 6.1.1), for $k \in [\![1, N_s - 1]\!]$;
- the orientation $o_k^s \in \mathcal{O}_\infty$ of the k^{th} segment can be deduced from the sequence of orientation changes (see Section 6.1.1), for $k \in [\![2, N_s]\!]$ (remind that the first orientation is $o_1^s = o^{ori.}$);
- the traversable cell $c_k^s \in \mathcal{C}$ to which the end point of the k^{th} segment belongs can be computed from the last interface crossed by segment k, for $k \in [\![1, N_s]\!]$.

Routing plan s		
N_s	:	Size
$\left[r_1^s, ..., r_{N_s-1}^s\right]$	:	Orientation changes
$\left[\mathcal{I}_1^s, ..., \mathcal{I}_{N_s}^s\right]$	:	Set of crossed interfaces
$Term^s$	:	$true/false$
$\left[b_1^s, ..., b_{N_s-1}^s\right]$	:	Bends
$\left[o_1^s, ..., o_{N_s}^s\right]$	:	Orientations
$\left[c_1^s, ..., c_{N_s}^s\right]$	:	Cells

(a) Description of plan s.



(b) Solution of LP_s^{CL}.

Figure 10.1 – A routing plan $s \in \mathcal{S}$.

10.1.2 Feasibility and cost-from-origin

The feasibility of a routing plan $s \in \mathcal{S}$ depends now on the possibility to create a neutral fibre using the selected bends and satisfying the space constraints defined by the interfaces to cross. This feasibility problem can be formulated by extending the lazy linear program LP_s^L introduced in Section 6.1.2 on page 72 into the *constrained lazy linear program* LP_s^{CL}. The new formulation contains four kinds of variables:

- *position variables* $p_k^s = \left(p_{k,x}^s, p_{k,y}^s, p_{k,z}^s\right)$ such that, for $k \in [\![1, N_s + 1]\!]$, real variable $p_{k,x}^s$ (respectively $p_{k,y}^s$ and $p_{k,z}^s$) is the x-coordinate (respectively y-coordinate and z-coordinate) of the k^{th} point of neutral fibre $\mathcal{F}_s$;
- *length variables* ℓ_k^s such that, for $k \in [\![1, N_s]\!]$, real variable ℓ_k^s is the length of the k^{th} segment of neutral fibre $\mathcal{F}_s$, or in other words $\ell_k^s = \left\|\overrightarrow{p_k^s p_{k+1}^s}\right\|$;
- *interface variables* $q_{k,i}^s = \left(q_{k,i,x}^s, q_{k,i,y}^s, q_{k,i,z}^s\right)$ such that, for $k \in [\![1, N_s]\!]$ and $i \in \mathcal{I}_k^s$, real variable $q_{k,i,x}^s$ (respectively $q_{k,i,y}^s$ and $q_{k,i,z}^s$) is the x-coordinate (respectively y-coordinate and z-coordinate) of the intersection between the k^{th} segment of neutral fibre $\mathcal{F}_s$ and an interface i it has to cross;
- *interface distance variables* $\alpha_{k,i}^s$ such that, for $k \in [\![1, N_s]\!]$ and $i \in \mathcal{I}_k^s$, real variable $\alpha_{k,i}^s$ is the distance between the k^{th} point of neutral fibre $\mathcal{F}_s$ and the intersection $q_{k,i}^s$ with the interface $i \in \mathcal{I}_k^s$ crossed by the k^{th} segment.

Linear program LP_s^{CL} still minimises the *cost-from-origin* of the neutral fibre part that must satisfy the constraints of plan s. Thus, linear program LP_s^{CL} can be formulated as follows:

$$\text{minimise } \sum_{k=1}^{N_s-1} \gamma_{b_k^s} + \mu \sum_{k=1}^{N_s} \ell_k^s \tag{10.1}$$

subject to:

$$p_1^s \in \mathcal{P}^{ori.} \tag{10.2}$$

$$p_{N_s+1}^s \in \mathcal{P}^{dest.} \quad \text{if } Term^s = true \tag{10.3}$$

$$p_{k+1}^s \in c_k^s \quad \forall k \in [\![1, N_s - 1]\!] \tag{10.4}$$

$$q_{k,i}^s \in \mathcal{P}_i \quad \forall k \in [\![1, N_s]\!] \quad \forall i \in \mathcal{I}_k^s \tag{10.5}$$

$$\ell_k^s \geq L_{b_{k-1}^s} + L^{min} + L_{b_k^s} \quad \forall k \in [\![1, N_s]\!] \tag{10.6}$$

$$\overrightarrow{p_k^s p_{k+1}^s} = \ell_k^s \overrightarrow{e_{o_k^s,z}} \quad \forall k \in [\![1, N_s]\!] \tag{10.7}$$

$$\overrightarrow{p_k^s q_{k,i}^s} = \alpha_{k,i}^s \overrightarrow{e_{o_k^s,z}} \quad \forall k \in [\![1, N_s]\!] \quad \forall i \in \mathcal{I}_k^s \tag{10.8}$$

$$\alpha_{k,i}^s \leq \ell_k^s \quad \forall k \in [\![1, N_s]\!] \quad \forall i \in \mathcal{I}_k^s \tag{10.9}$$

$$\ell_k^s \in \mathbb{R}^+ \quad \forall k \in [\![1, N_s]\!] \tag{10.10}$$

$$\alpha_{k,i}^s \in \mathbb{R}^+ \quad \forall k \in [\![1, N_s]\!] \quad \forall i \in \mathcal{I}_k^s \tag{10.11}$$

$$p_k^s \in \mathbb{R}^3 \quad \forall k \in [\![1, N_s + 1]\!] \tag{10.12}$$

$$q_{k,i}^s \in \mathbb{R}^3 \quad \forall k \in [\![1, N_s]\!] \quad \forall i \in \mathcal{I}_k^s \tag{10.13}$$

Most constraints are similar to lazy linear program LP_s^L introduced in Section 6.1.2. There are four new kinds of constraints here. Constraints 10.4 force the successive break points of neutral fibre $\mathcal{F}_s$ to belong to the traversable cell to which they are allocated given plan s. Constraints 10.5, 10.8 and 10.9 impose that the intersection between the k^{th} segment and an interface i it has to cross must belong both to the interface and to the segment (see Figure 10.2).

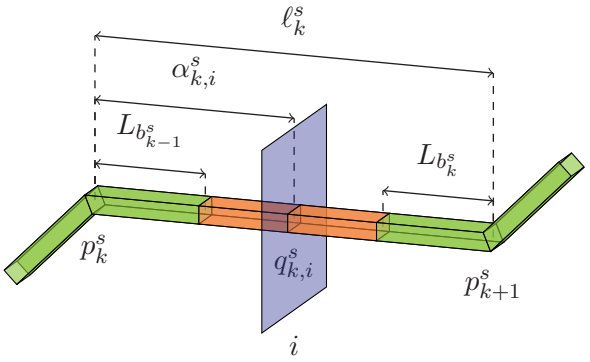


Figure 10.2 – Portion of a pipe between two successive break points of the neutral fibre (the straight section with variable length is depicted in orange).

If LP_s^{CL} has a solution, then routing plan s is *feasible*. The optimal cost-from-origin of linear program LP_s^{CL} is called *constrained lazy cost* and is referred to as $g_{CL}(s)$. The corresponding waveguide can be rebuilt with Algorithm 10 on page 59. On the contrary, when there is no solution, routing plan s is not feasible and, by convention, its minimal cost-from-origin is infinite, that is to say $g_{CL}(s) = \infty$.

By contrast with the lazy linear program LP_s^L introduced for the FWRP, the constrained lazy feasibility problem LP_s^{CL} can be infeasible because it can be impossible to cross an interface with a given bend combination. Therefore, there is no obvious solution to constrained lazy feasibility problem LP_s^{CL} for the CWRP because of space constraints.

10.1.3 Neighbourhood

A feasible and non-terminated routing plan $s \in \mathcal{S}$ in constrained space can be extended using three kinds of decisions: add a bend, cross an interface or finish the plan. So, successors are built by forward chaining. In cases where plan s is infeasible, it cannot be extended.

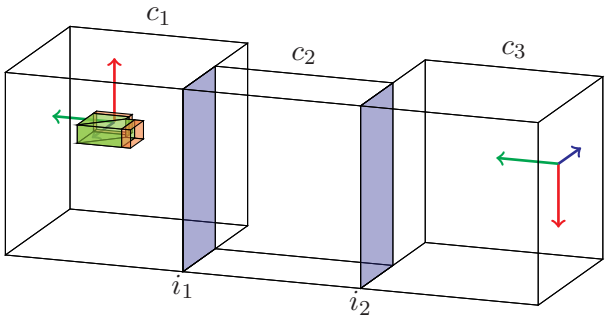
Remind that there is a single way to reach a routing plan $s \in \mathcal{S}$ which is to apply exactly the same sequence of bends and to cross the same sequence of interfaces, and then terminate the plan if needed. So, it is impossible to visit a plan s more than once.

Add a bend

If the maximum number of bends $N_S - 1$ is not reached, that means $N_s < N_S$, and if $r \in \mathcal{R}_\infty^{out.}\left(o_{N_s}^s\right)$ is a reachable orientation change such that $N_s + minBends\left(o_r^+\right) \leq N_S$ (the maximum number of segments is not exceeded), then it is possible to apply bend b_r at the end of the last segment of routing plan s. Let $s' \in \mathcal{S}$ be the successor resulting from this decision. Formally, adding bend b_r associated with orientation change r creates a new segment on the neutral fibre $\mathcal{F}_s$, that is to say $N_{s'} = N_s + 1$, and extends the sequence of orientation changes with r, meaning that $r_k^{s'} = r_k^s$ for $k \in [\![1, N_s - 1]\!]$ and $r_{N_s}^{s'} = r$. Then, the sets of interfaces crossed by each segment remain unchanged, meaning that $\mathcal{I}_k^{s'} = \mathcal{I}_k^s$ for $k \in [\![1, N_s]\!]$. Of course, the new segment does not cross any interface, so $\mathcal{I}_{N_{s'}}^{s'} = []$. The successor s' is described on Figure 10.3a and illustrated on Figure 10.3b.

Routing plan s'
$N_{s'} = N_s + 1$
$\left[r_1^s, ..., r_{N_s-1}^s, r\right]$
$\left[\mathcal{I}_1^s, ..., \mathcal{I}_{N_s}^s, []\right]$
$false$
$\left[b_1^s, ..., b_{N_s-1}^s, b_r\right]$
$\left[o_1^s, ..., o_{N_s}^s, o_r^+\right]$
$\left[c_1^s, ..., c_{N_s}^s, c_{N_s}^s\right]$

(a) Description of successor s'.



(b) Solution of $LP_{s'}^{CL}$.

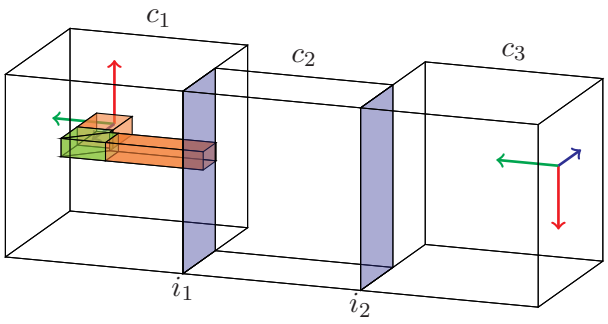
Figure 10.3 – Adding a bend associated with an orientation change $r \in \mathcal{R}_\infty^{out.}\left(o_{N_s}^s\right)$ to a plan $s \in \mathcal{S}$.

Cross an interface

If $i \in \mathcal{I}_{c_{N_s}^s}^{out.}$ is a crossable interface with a normal $\overrightarrow{n_i}$ that has a positive dot product $\overrightarrow{n_i} \cdot \overrightarrow{e_{o_{N_s}^s,z}} \geq 0$ with the orientation $o_{N_s}^s$ of the last segment of the neutral fibre $\mathcal{F}_s$, then it is possible to add interface i to the sequence of interfaces crossed by this last segment. Let $s' \in \mathcal{S}$ be the successor resulting from this decision. Formally, crossing interface i does not modify the number of segments in neutral fibre $\mathcal{F}_s$ nor the sequence of orientation changes, meaning that $r_k^{s'} = r_k^s$ for $k \in [\![1, N_s - 1]\!]$. Then, the sets of interfaces crossed by the $N_s - 1$ first segments remain unchanged while interface i is added to the sequence of interfaces crossed by the last segment, meaning that $\mathcal{I}_k^{s'} = \mathcal{I}_k^s$ for $k \in [\![1, N_s - 1]\!]$ and $\mathcal{I}_{N_s}^{s'} = \mathcal{I}_{N_s}^s \cup \{i\}$. As a consequence, the last vertex of neutral fibre $\mathcal{F}_s$ is allocated in the new traversable cell c_i^+. The successor s' is described on Figure 10.4a and illustrated on Figure 10.4b on the facing page.

Routing plan s'
$N_{s'} = N_s$
$\left[r^s_1, ..., r^s_{N_s-1}\right]$
$\left[\mathcal{I}^s_1, ..., \mathcal{I}^s_{N_s-1}, \mathcal{I}^s_{N_s} \cup [i]\right]$
$false$
$\left[b^s_1, ..., b^s_{N_s-1}\right]$
$\left[o^s_1, ..., o^s_{N_s}\right]$
$\left[c^s_1, ..., c^s_{N_s-1}, c^+_i\right]$

(a) Description of successor s'.



(b) Solution of $LP^{CL}_{s'}$.

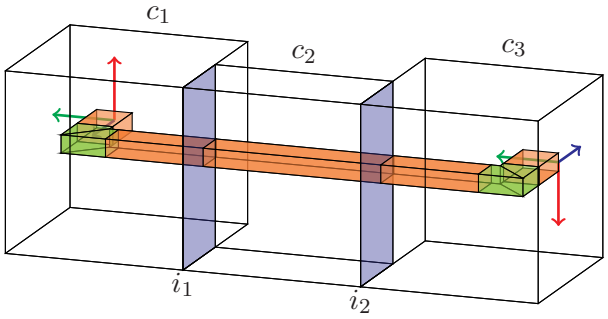
Figure 10.4 – Crossing an interface $i \in \mathcal{I}^{out.}_{c^s_{N_s}}$ from a plan $s \in \mathcal{S}$.

Finish a plan

If a destination orientation and a traversable cell have been reached, that is to say $o^s_{N_s} \in \mathcal{O}^{dest.}$ and $c^s_{N_s} \in \mathcal{C}^{dest.}$, then routing plan s can be terminated. Its successor s' is the same as plan s but it is terminated, meaning that $Term^s = true$, and it has to reach the destination polyhedron $\mathcal{P}^{dest.}$ (see Constraint 10.3).

Routing plan s'
$N_{s'} = N_s$
$\left[r^s_1, ..., r^s_{N_s-1}\right]$
$\left[\mathcal{I}^s_1, ..., \mathcal{I}^s_{N_s}\right]$
$true$
$\left[b^s_1, ..., b^s_{N_s-1}\right]$
$\left[o^s_1, ..., o^s_{N_s}\right]$
$\left[c^s_1, ..., c^s_{N_s}\right]$

(a) Description of successor s'.



(b) Solution of $LP^{CL}_{s'}$.

Figure 10.5 – Terminating a plan $s \in \mathcal{S}$.

10.1.4 Origin and destination plans

The aim is to find a destination plan that is a feasible and terminated routing plan $s \in \mathcal{S}$ for which the last orientation is a destination orientation and the last cell is a destination traversable cell, that means $o^s_{N_s} \in \mathcal{O}^{dest.}$ and $c^s_{N_s} \in \mathcal{C}^{dest.}$ (see Figure 10.6b). Like for the FWRP, the set of destination plans is referred to as $\mathcal{S}_{dest.}$. The exploration of the space of routing plans $\mathcal{S}$ starts from the initial plan, referred to as $s_{ori.}$, which has the origin orientation $o^{ori.}$, starts in the origin cell $c^{ori.}$, does not contain any bend and does not cross any interface, that means $N_{s_{ori.}} = 1$ and $\mathcal{I}^{s_{ori.}}_1 = \{\}$, as shown on Figure 10.6a. Of course, it is a non-terminated plan.

Routing plan $s_{ori.}$
$N_s = 1$
$\varnothing$
$[\{\}]$
$false$
$\varnothing$
$\left[o^{ori.}\right]$
$\left[c^{ori.}\right]$

(a) Origin routing plan $s_{ori.}$.

Routing plan s
$N_s \leq N_S$
$\left[r^s_1, ..., r^s_{N_s-1}\right]$
$\left[\mathcal{I}^s_1, ..., \mathcal{I}^s_{N_s}\right]$
$true$
$\left[b^s_1, ..., b^s_{N_s-1}\right]$
$\left[o^s_1, ..., o^s_{N_s}\right]$ with $o^s_{N_s} \in \mathcal{O}^{dest.}$
$\left[c^s_1, ..., c^s_{N_s}\right]$ with $c^s_{N_s} \in \mathcal{C}^{dest.}$

(b) Destination routing plan $s \in \mathcal{S}_{dest.}$.

Figure 10.6 – Origin and destination routing plans.

10.2 Evaluations and heuristics

In order to choose the most promising routing plans when using ISAs, a heuristic function $h(s)$ is necessary to estimate the minimal cost required to extend $s \in \mathcal{S}$ and reach a feasible and terminated destination plan in $\mathcal{S}_{dest.}$. This section proposes three heuristic functions which use the neutral fibre provided by the constrained lazy linear program LP^{CL}_s. The first naive one consist in using the as the crow flies distance to the destination (see Section 10.2.1). The two other heuristics, respectively presented in Section 10.2.3 and Section 10.2.4, are more precise since they take space constraints into account using the notions of *candidate trails* and *extended plans* introduced in Section 10.2.2.

10.2.1 Distance as the crow flies

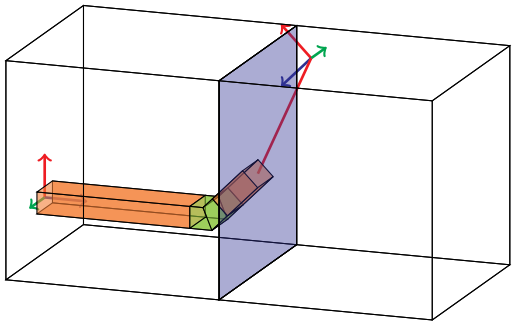
A first heuristic evaluation of a routing plan $s \in \mathcal{S}$ is the distance as the crow flies between the last point $p^s_{N_s+1}$ and the destination polyhedron $\mathcal{P}^{dest.}$ weighted with linear cost μ, referred to as $h_{ACF}(s)$. It is a naive heuristic since it does not take into account the routing space constraints.

Definition 29: As the crow flies heuristic h_{ACF}

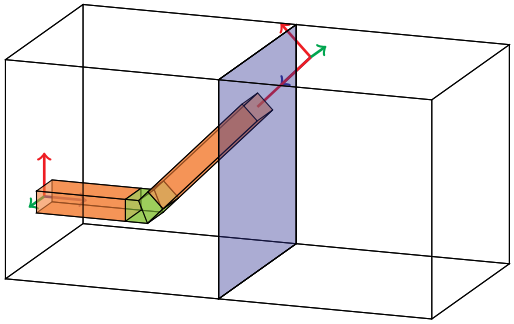
Let s be a feasible routing plan in $\mathcal{S}$ and $\left[p^s_1, ..., p^s_{N_s+1}\right]$ the polyline provided by constrained lazy linear program LP^{CL}_s which minimises the cost-from-origin. The *as the crow flies heuristic* is defined for routing plan s by:

$$h_{ACF}(s) = \begin{cases} \gamma_{min.}\left(o^s_{N_s}\right) + \mu \min_{p \in \mathcal{P}^{dest.}} \left\|\overrightarrow{p^s_{N_s+1}p}\right\| & \text{if } Term^s = false \\ 0 & \text{otherwise} \end{cases}$$

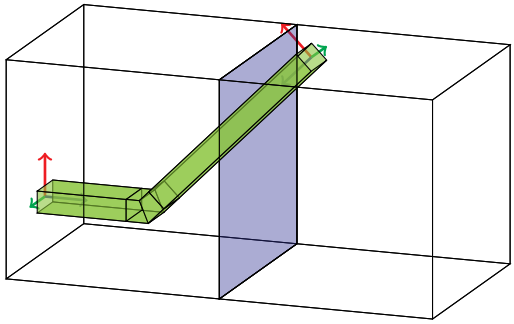
By contrast with lazy heuristic h_L introduced to solve the FWRP, the as the crow flies heuristic h_{ACF} is not admissible. Figure 10.7 on the next page shows a counter-example in which a routing plan $s \in \mathcal{S}$, resulting from the addition of a 45°-bend and the crossing of an interface, is evaluated with constrained lazy linear program LP^{CL}_s. Figure 10.7a presents an optimal solution of LP^{CL}_s and the estimation $h_{ACF}(s)$ of the as the crow flies distance to reach the destination. However, as illustrated on Figure 10.7b, it is possible to build a sub-optimal solution for LP^{CL}_s with a lower estimation $h_{ACF}(s)$. This sub-optimal solution can match an optimal neutral fibre $\mathcal{F}_s$ that reaches the destination polyhedron $\mathcal{P}^{dest.}$, as shown on Figure 10.7c.



(a) Optimal solution for linear program LP_s^{CL}.



(b) Sub-optimal solution for linear program LP_s^{CL}.



(c) Optimal solution of the instance.

Figure 10.7 – Non admissibility of the as the crow flies heuristic h_{ACF} in constrained space.

10.2.2 Space of candidate trails

To get more accurate estimations of the distance to the destination, the remaining part of the neutral fibre $\mathcal{F}_s$, that is not constrained to use bends from the catalogue, must consider the space constraints and avoid obstacles rather than using a straight segment to reach the destination polyhedron $\mathcal{P}^{dest.}$. Ideally, this polyline to the destination must be as short as possible. Finding such a polyline corresponds to the Euclidean Shortest Path Problem (ESPP).

Euclidean Shortest Path Problem

The ESPP in two-dimensional space can be solved in polynomial time. A classical approach consists in using shortest path algorithms in a visibility graph that has been precomputed from the obstacles [60, 51, 52]. Is is also possible to propagate a frontier from the source point until it meets the target point [39]. A simplified version of the 2D-ESPP searches for the geometric shortest path in a grid that can contain obstacle cells. This case can be solved efficiently with the *Theta** algorithm proposed by NASH [67] which is a variant of A* propagating information along grid edges without constraining the paths to follow these edges. More recently, an interval-based search technique, called *Anya*, has been introduced in [36] and extended to support search over arbitrary sets of convex polygons [22].

In a three-dimensional space, the ESPP has been proved to be a NP-hard problem [14]. In this case, the shortest path does not necessarily pass through vertices of the obstacles but it is shown to pass through their edges [86]. Approximation algorithms have been proposed to solve the 3D-ESPP in polynomial time. The main idea is to sample points along the edges of the obstacles in order to build a visibility graph [70, 18]. Then, as for the 2D-ESPP, a shortest path algorithm is applied (typically DIKJSTRA's algorithm). Furthermore, it can be noticed that the *Theta** algorithm can be generalised to three-dimensional space by considering a 3D-grid [68].

Trail

A way of considering the space constraints in the estimation of the remaining distance, inspired from existing ideas to solve the 3D-ESPP, is to build so-called *candidate trails* between some specific points of the routing space. Formally, a *trail* for a routing plan $s \in \mathcal{S}$ is a polyline inside the routing space that connects the end point $p^s_{N_s+1}$ of s provided by LP^{CL}_s to the destination polyhedron $\mathcal{P}^{dest.}$.

Definition 30: Trail

For a routing plan $s \in \mathcal{S}$, a *trail* t is a polyline $\left[m^t_1, ..., m^t_{N_t+1}\right]$ which connects the last point $p^s_{N_s+1}$ to the destination polyhedron $\mathcal{P}^{dest.}$ and stays inside the routing space $G(\mathcal{C}, \mathcal{I})$ (see Figure 10.8), what can be translated as follows:

$$
\begin{aligned}
m^t_1 &= p^s_{N_s+1} \\
m^t_{N_t+1} &\in \mathcal{P}^{dest.} \\
\left[m^t_k, m^t_{k+1}\right] &\subset \bigcup_{c \in \mathcal{C}} \mathcal{P}_c \quad \forall k \in [\![1, N_t]\!]
\end{aligned}
$$

The set of trails for a routing plan $s \in \mathcal{S}$ is referred to as $\mathcal{T}_s$.

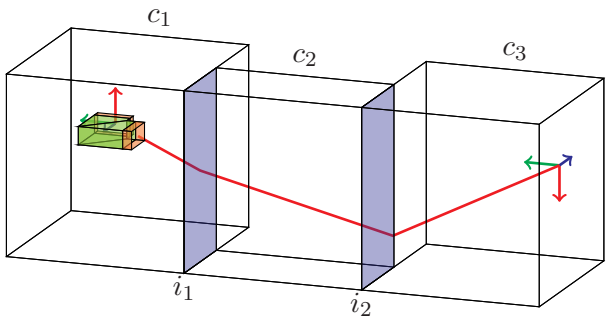


Figure 10.8 – An example of trail (depicted in red).

In a trail, the constraints expressing that each orientation change of the polyline must correspond to a bend of the catalogue are ignored, the goal being only to estimate the quality of a routing plan. Furthermore, since a trail $t \in \mathcal{T}_s$ stays inside the routing space, it is possible to define the sequence of interfaces, called *channel*, that are crossed by trail t. In what follows, K_t refers to the number of interfaces crossed by trail t and i^t_k is the k^{th} interface crossed by trail t.

Generation of candidate trails

To generate candidate trails, a graph $G(\mathcal{M}, \mathcal{D})$ called *trail space* is built during a preprocessing step by sampling a set of points $\mathcal{M}(i)$ on each interface $i \in \mathcal{I}$ using BRIDSON's algorithm [11]. It is a maximal POISSON-disk sampling method which ensures that sampled points are separated by a *sampling radius* $\rho \in \mathbb{R}^+$, such that when ρ decreases, the density of sampled points increases (see Figure 10.9). For a point $m \in \bigcup_{i \in \mathcal{I}} \mathcal{M}(i)$, the interface on which point m has been sampled is referred to as i_m. A set of points

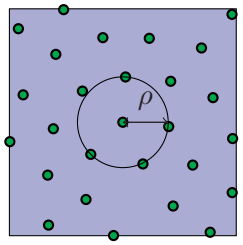


Figure 10.9 – Maximal POISSON-disk sampling with a radius $\rho \in \mathbb{R}^+$.

$\mathcal{M}\left(\mathcal{P}^{dest.}\right)$ is also generated in the destination polyhedron $\mathcal{P}^{dest.}$ and is joined to the sets of points sampled on interfaces to form the set of nodes $\mathcal{M}$ of the trail space. Then, the edges of graph $G\left(\mathcal{M},\mathcal{D}\right)$ are built using Algorithm 12. The idea is to propagate a frontier from the points sampled in the destination polyhedron in order to evaluate the distance L_m from each point m sampled on an interface $i \in \mathcal{I}$ to the destination by jumping from a sampled point to another on a neighbouring interface. Two distinct interfaces $i, i' \in \mathcal{I}$ are considered as neighbours if they involve a common traversable cell c, that is to say $i, i' \in \mathcal{I}(c)$. Each point m is connected to the point m' of a neighbour interface $i' \in \mathcal{I}$ that has the minimal distance $L_{m'}$ to the destination. To do so, Algorithm 12 follows a procedure similar to the DIKJSTRA's algorithm. Of course, a point $m \in \mathcal{M}\left(\mathcal{P}^{dest.}\right)$ sampled in the destination polyhedron satisfies $L_m = 0$. An example of trail space $G\left(\mathcal{M},\mathcal{D}\right)$ is illustrated on Figure 10.10.

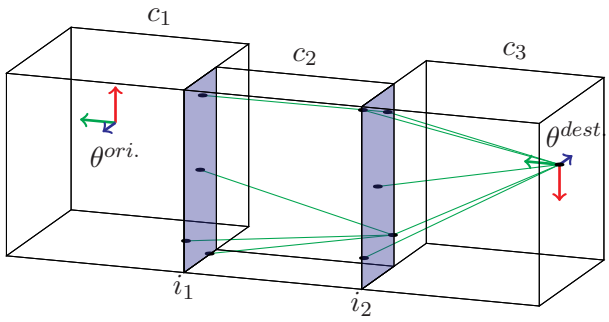


Figure 10.10 – Example of $G\left(\mathcal{M},\mathcal{D}\right)$.

It is important to note that the candidate trails presented here are not guaranteed to be the 3D-shortest paths to the destination polyhedron $\mathcal{P}^{dest.}$ because of the sampling approach used, even if reducing sampling radius ρ allows getting closer to optimal paths. As a consequence, the two heuristics introduced in what follows are not necessarily admissible and can clearly overestimate the real cost required to reach the goal configuration. This remark implies that the first feasible and terminated routing plan reached by A*-like algorithms with these heuristics is not necessarily optimal.

10.2.3 Trail length heuristic and extended routing plan

Instead of using the as the crow flies distance, a better estimation of the remaining distance to reach the destination polyedron $\mathcal{P}^{dest.}$ from a routing plan $s \in \mathcal{S}$ is the length of the shortest trail $t^*\left(s\right)$ that can be found in $G\left(\mathcal{M},\mathcal{D}\right)$, as illustrated on Figure 10.11.

Definition 31: Shortest trail $t^*\left(s\right)$

Let s be a feasible routing plan in $\mathcal{S}$. The *shortest trail* $t^*\left(s\right)$ for routing plan s is the trail passing through the node $m^*\left(s\right)$ that minimises the trail length among the nodes sampled on the interfaces of the last cell reached by plan s, what can be written as follows:

$$m^*\left(s\right) = \underset{\substack{m \in \mathcal{M}(i) \\ i \in \mathcal{I}^{out.}_{c^s_{N_s+1}}}}{\arg\min} \left(\left\| \overrightarrow{p^s_{N_s+1} m} \right\| + L_m \right)$$

The estimation can be further improved by trying to get closer to the destination in order to reach it faster. To do so, it is possible to use the shortest trail $\tilde{t}$ for an *extended routing*

Algorithm 12: Generate the trail space $G(\mathcal{M}, \mathcal{D})$

Input:

- Set of destination cells: $\mathcal{C}^{dest.}$
- Set of points sampled in destination polyhedron: $\mathcal{M}\left(\mathcal{P}^{dest.}\right)$
- Sets of points sampled on each interface: $\mathcal{M}(i) \qquad \forall i \in \mathcal{I}(c)$

for $m \in \mathcal{M}\left(\mathcal{P}^{dest.}\right)$ **do**
  $L_m \leftarrow 0$
  Add node m in $\mathcal{M}$
$OpenList \leftarrow \varnothing$
for $c \in \mathcal{C}^{dest.}$ **do**
  for $i \in \mathcal{I}_c^{inc.}$ **do**
    for $m \in \mathcal{M}(i)$ **do**
      if m *was not visited before* **then**
        $L_m \leftarrow \infty$
      for $m' \in \mathcal{M}\left(\mathcal{P}^{dest.}\right)$ **do**
        if $\left\|\overrightarrow{m'm}\right\| < L_m$ **then**
          $L_m \leftarrow \left\|\overrightarrow{m'm}\right\|$
          $bestParent(m) \leftarrow m'$
      Add node m in $OpenList$ with value L_m
while $OpenList \neq \varnothing$ **do**
  Remove m with the smallest L_m-value from $OpenList$
  if $m \notin \mathcal{M}$ **then**
    Add node m in $\mathcal{M}$
  for $i \in \mathcal{I}_{c_{i_m}^-}^{inc.}$ **do**
    for $m' \in \mathcal{M}(i)$ **do**
      if m' *was not visited before* **then**
        $L_{m'} \leftarrow \infty$
      if $\left\|\overrightarrow{mm'}\right\| < L_{m'}$ **then**
        $L_{m'} \leftarrow \left\|\overrightarrow{mm'}\right\|$
        $bestParent(m') \leftarrow m$
        Add node m' in $OpenList$ with value $L_{m'}$
for $m \in \bigcup_{i \in \mathcal{I}} \mathcal{M}(i)$ **do**
  Add arc $(m, bestParent(m))$ in $\mathcal{D}$
return $G(\mathcal{M}, \mathcal{D})$

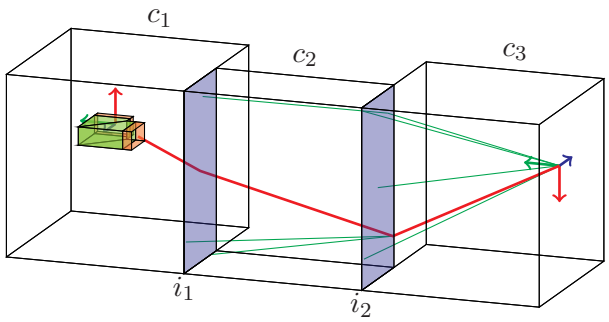


Figure 10.11 – Trail heuristic on a simple routing case (candidate trails are depicted in green and the shortest one in red).

plan $\tilde{s} \in \mathcal{S}$ obtained from plan s by crossing as many interfaces as possible along the channel of shortest trail $t^*(s)$ from plan s without adding any bend. The extended plan $\tilde{s}$ of plan s can be computed using Algorithm 13. The feasibility of the extended plans is evaluated using the linear programs $LP_{\tilde{s}}^{CL}$, whose solutions can be reused when the extended plans are expanded in turn. Finally, the resulting heuristic can be defined as follows.

Definition 32: Trail length heuristic h_{TL}

Let s be a feasible routing plan in $\mathcal{S}$ and $\tilde{t}$ its extended plan. Let $\tilde{t}$ be the shortest trail for extended plan $\tilde{s}$ and $\left[p_1^{\tilde{s}}, ..., p_{N_{\tilde{s}}+1}^{\tilde{s}}\right]$ the polyline provided by constrained lazy linear program $LP_{\tilde{s}}^{CL}$ which minimises the cost-from-origin. The *trail length heuristic* is defined for routing plan s by:

$$h_{TL}(s) = \begin{cases} \gamma_{min.}\left(o_{N_{\tilde{s}}}^{\tilde{s}}\right) + \mu \sum_{k=1}^{N_{\tilde{t}}} \left\| \overrightarrow{m_k^{\tilde{t}} m_{k+1}^{\tilde{t}}} \right\| & \text{if } Term^{\tilde{s}} = false \\ 0 & \text{otherwise} \end{cases}$$

Algorithm 13: Compute the extended plan $\tilde{s}$

Input:

- Routing plan: s
- Shortest trail: $t^*(s)$

$\tilde{s} \leftarrow s$
for $k \in [\![1, K_{t^*(s)}]\!]$ **do**
  Compute plan s' resulting from crossing interface i_k^t from plan $\tilde{s}$
  if s' *in feasible* **then**
    $\tilde{s} \leftarrow s'$
  else
    return $\tilde{s}$
Compute plan s' resulting from terminating plan $\tilde{s}$
if s' *in feasible* **then**
  $\tilde{s} \leftarrow s'$
return $\tilde{s}$

10.2.4 Trail cost heuristic

Previous heuristics only estimate the minimal bend combination cost $\gamma_{min.}(o)$ to reach a destination orientation and the lineic contribution to the cost of a waveguide. However, the polyline $\left[m_1^{\tilde{t}}, ..., m_{N_{\tilde{t}}+1}^{\tilde{t}}\right]$ of shortest trail $\tilde{t}$ for extended plan $\tilde{s}$ can also be used to estimate a more accurate cost of the remaining bends that will be added to the waveguide to reach the destination while avoiding obstacles (see Figure 10.12). This approach is described in Algorithm 14, that returns an estimated cost $\gamma_{est.}(o,t)$ of the best bend combination to follow trail t starting from orientation $o \in \mathcal{O}$. This algorithm analyses the successive orientation changes on the trail and tries to reproduce them with bends from the catalogue. It estimates the bend cost using a function σ that takes as an input the orientation o of the waveguide and a vector $\overrightarrow{u}$, and that returns a quantity $\sigma(o, \overrightarrow{u}) = \overrightarrow{e_{o,z}} \cdot \frac{\overrightarrow{u}}{\|\overrightarrow{u}\|}$ estimating through a scalar product the angular proximity between the main direction $\overrightarrow{e_{o,z}}$ of orientation o and the direction defined by $\overrightarrow{u}$. The closer $\sigma(o, \overrightarrow{u})$ is to 1, the closer o is from direction $\overrightarrow{u}$. Note that with such an approach, the orientation of the waveguide section around the neutral fibre is ignored. Moreover, from the last orientation o reached when following the trail, Algorithm 14 also adds the cost $\gamma_{min.}(o)$ of the minimal bend combination that reaches a destination orientation from orientation o. Remind that this cost is precomputed for any reachable orientation o, before using Algorithm 14.

Algorithm 14: Evaluate $\gamma_{est.}(o,t)$

Input:

- Orientation: o
- Trail: t

$o' \leftarrow o$
$\gamma_{est.}(o,t) \leftarrow 0$
for $k \in [\![1, N_t]\!]$ **do**
  do
    $improvement \leftarrow false$
    $b^* \leftarrow \text{argmax}_{b \in B_{cat.}} \left(\sigma\left(rot_{M_b}(o), \overrightarrow{m_k^t m_{k+1}^t}\right)\right)$
    if $\sigma\left(rot_{M_{b^*}}(o), \overrightarrow{m_k^t m_{k+1}^t}\right) > \sigma\left(o, \overrightarrow{m_k^t m_{k+1}^t}\right)$ **then**
      $improvement \leftarrow true$
      $\gamma_{est.}(o,t) \leftarrow \gamma_{est.}(o,t) + \gamma_{b^*}$
      $o \leftarrow rot_{M_{b^*}}(o)$
  while $improvement = false$
$\gamma_{est.}(o,t) \leftarrow \gamma_{est.}(o,t) + \gamma_{min.}(o)$
return $\gamma_{est.}(o,t)$

From the estimated cost of the bend combination required to reach the destination avoiding obstacles, it is possible to build a more accurate trail cost heuristic.

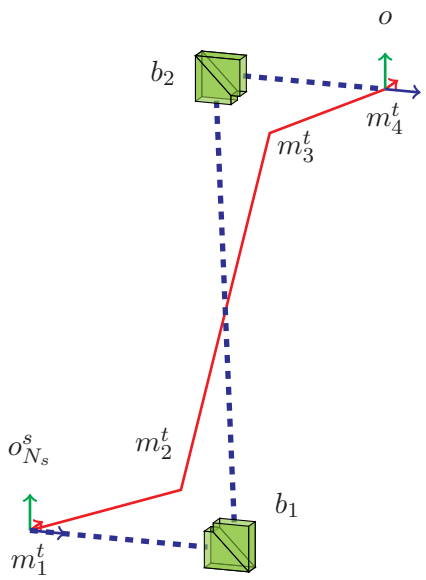


Figure 10.12 – Estimation of the bend combination needed to follow a trail $t \in \mathcal{T}(s)$ from plan $s \in \mathcal{S}$.

Definition 33: Trail cost heuristic h_{TC}

Let s be a feasible routing plan in $\mathcal{S}$ and $\tilde{s}$ its extended plan. Let $\tilde{t}$ be the shortest trail for extended plan $\tilde{s}$ and $\left[p_1^{\tilde{s}}, ..., p_{N_{\tilde{s}}+1}^{\tilde{s}}\right]$ the polyline provided by constrained lazy linear program $LP_{\tilde{s}}^{CL}$ which minimises the cost-from-origin. The *trail cost heuristic* is defined for routing plan s by:

$$h_{TC}(s) = \begin{cases} \gamma_{est.}\left(o_{N_{\tilde{s}}}^{\tilde{s}}, \tilde{t}\right) + \mu \sum_{k=1}^{N_{\tilde{t}}} \left\| \overrightarrow{m_k^{\tilde{t}} m_{k+1}^{\tilde{t}}} \right\| & \text{if } Term^{\tilde{s}} = false \\ 0 & \text{otherwise} \end{cases}$$

10.3 Experiments on the CWRP

The three proposed heuristics, as the crow flies h_{ACF}, trail length h_{TL}, and trail cost h_{TC}, have been implemented in Java with A*, WA*, Greedy BFS, SAHC and BrF-BS. Remind that the implementation of these algorithms does not require to maintain any closed list of the already visited states. The Simplex solver of the Apache Commons Math library (version 3.6.1) [3] is still used to solve the linear programs. In the results that follow, the search has been stopped as soon as a solution was found and the runtime has been limited to 3 minutes, which corresponds to the maximal acceptable time to solve the instances in an industrial context.

First, test instances which are more complex than the ones used for the MILP formulation are presented in Section 10.3.1. Then, the performances of each heuristic are evaluated using A*, WA* and Greedy BFS in order to select the best parameters to solve the CWRP (see Section 10.3.2). The results obtained using SAHC and BrF-BS are not reported here because of their poor success rate on the test instances. Last, the impact of the trail space on the resolution speed and on the solution quality with the trail cost heuristic is studied in Section 10.3.3.

10.3.1 Instance sets

As previously, the formulation of the CWRP adapted to ISAs has been experimented on three instance sets corresponding to bend catalogues $B_{cat.}^{90^\circ}$, $B_{cat.}^{45^\circ}$ and $B_{cat.}^{30^\circ}$ (see Table 11 on page 170). For each bend catalogue $B_{cat.}$, the instance set contains ten instances with gradual difficulty built as follows. The first instance uses the routing space with the traversable cells described in Table 7 and the origin and destination configurations detailed in Table 9 on page 168. Note that it is identical to instance 4 used for the MILP formulation of the CWRP. Then, other instances are derived from the first one by taking an increasing number of obstacles into account, which leads to subdivide traversable cells and results in routing spaces with more cells. The obstacles considered by each instance are defined in Table 10 on page 169. The ten instances are illustrated on Figure 10.13 on page 138. All other features, like the linear and bend costs, are the same as the ones used in Section 9.3.

10.3.2 Tuning the Best-First Search

The WA* algorithm has been studied for the following values of weight ϵ: 1.1, 1.2, 1.5, 2, 3, 5 and 8. A*, which is WA* with $\epsilon = 1$, and Greedy BFS, which can be seen as WA* with $\epsilon = \infty$, are also tested. Figure 10.14 on page 139, Figure 10.15 on page 140 and Figure 10.16 on page 141 present the results on the three instance sets using respectively the as the crow flies, trail length, and trail cost approaches with a radius $\rho = 10$ used for sampling points on interfaces. Note that the runtimes reported here include the construction of the trail space and the resolution for the trail heuristics.

As expected, approach $f_{ACF} = g_{CL} + \epsilon h_{ACF}$ which uses the distance as the crow flies does not succeed in solving instances with the largest catalogues $B_{cat.}^{45^\circ}$ and $B_{cat.}^{30^\circ}$. Only the first instances using the simple catalogue $B_{cat.}^{90^\circ}$ can be solved within the available time with large ϵ-values like $\epsilon = 3$ and $\epsilon = 5$ (see Figure 10.14). This confirms that such a naive approach which does not consider the topology of the routing space cannot be used on industrial instances.

By contrast, approximating the distance to the destination using the length of the shortest trail with heuristic h_{TL} allows to solve larger instances with more traversable cells and more complex bend catalogues, as shown on Figure 10.15. Nevertheless, very few instances are solved with the industrial catalogue $B_{cat.}^{30^\circ}$. When looking at solution quality, it seems that the first solution encountered by WA* becomes more expensive when ϵ increases (see Figure 10.15b). Here, WA* provides the best performances on the three instance sets with $\epsilon = 2$ and 3.

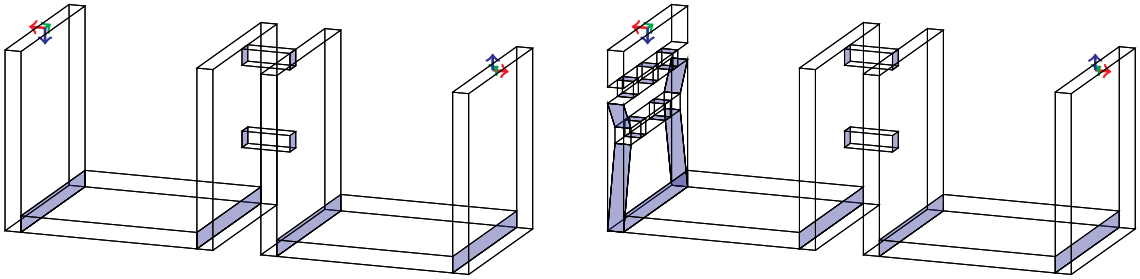
Trail cost heuristic h_{TC} outperforms the previous ones when the number of cells increases, that is to say when there are more obstacles to avoid (see Figure 10.16). Indeed, it nearly solves all instances within 3 minutes using catalogues $B_{cat.}^{90^\circ}$ and $B_{cat.}^{45^\circ}$, even with small values of ϵ. Moreover, several ϵ-values provide solutions for half of the instances with the industrial catalogue $B_{cat.}^{30^\circ}$. Similarly to heuristic h_{TC}, a trend visible on Figure 10.16b shows that smaller ϵ-values allow to find better solutions. However, more routing plans are expanded in comparison with large ϵ-values, and as a consequence the resolution takes longer. A good trade-off between resolution speed and solution quality consists in choosing $\epsilon = 2$ or 3 with this heuristic.

Figure 10.17 on page 142 compares the three heuristics using the best ϵ-values. In particular, Figure 10.17c illustrates that heuristic h_{TC} expands fewer plans than heuristic h_{TL} which, in turn, expands fewer plans than heuristic h_{ACF} by an order of magnitude. This confirms the efficiency of the successive improvements made with the trail approach. A detailed analysis of routing plans expanded by the different heuristics shows that considering only the cost $\gamma_{min.}\left(o_{N_s}^s\right)$ of the minimal bend combination to reach a destination orientation in $\mathcal{O}^{dest.}$, like heuristic h_{ACF} and h_{TL}, misleads the exploration of the space of routing plans. Indeed, it favours plans that apply bends to reach a destination orientation as fast as possible rather than

moving towards a destination cell in $\mathcal{C}^{dest.}$. It is all the more accentuated when bends have large costs in comparison with the linear cost μ.

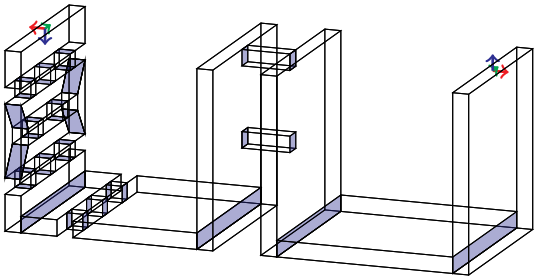
Finally, it appears that trail cost heuristic h_{TC} is clearly more robust to increasing the bend catalogue and routing space sizes, even if trail length heuristic h_{TL} tends to provide better solutions for the same value of ϵ. This better solution quality may be explained by the fact that the overestimation of heuristic h_{TC} is larger than the one obtained with the trail length heuristic since h_{TC} approximates the remaining bend cost by generating a sub-optimal bend combination to follow the shortest trail. Nevertheless, the solutions found by the trail cost heuristic are competitive and completely acceptable for a waveguide designer, as illustrated on Figure 10.18.

Obviously, the regularity of the solutions depends on the bend costs. The instances used in this thesis strongly penalise the bends in comparison with the linear cost, which favors the regularity of the waveguides. Bend costs closer to the linear cost would lead to less regular optimal waveguides and the approach would provide solutions that might be rejected by designers because of their high number of bends.

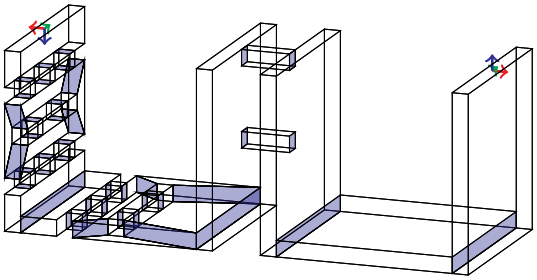


(a) Instance 1.

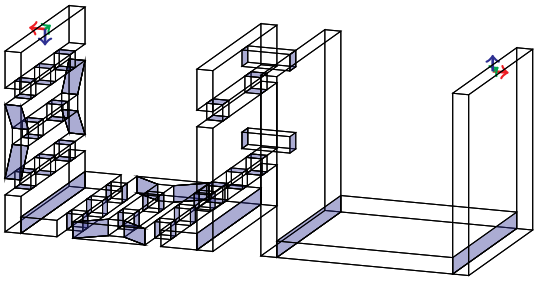
(b) Instance 2.



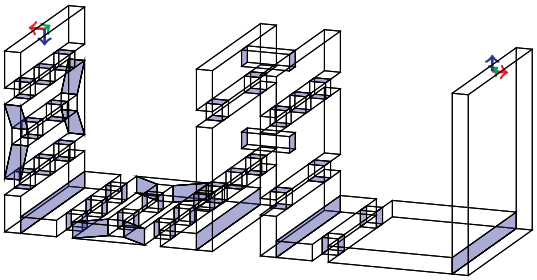
(c) Instance 3.



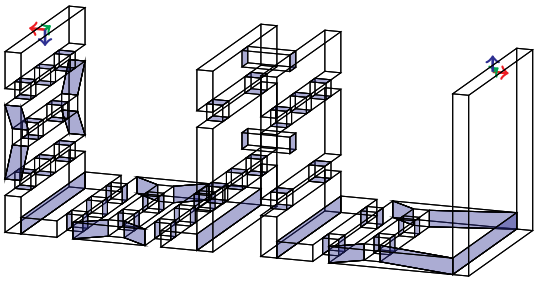
(d) Instance 4.



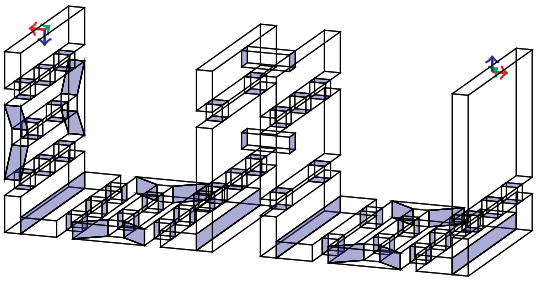
(e) Instance 5.



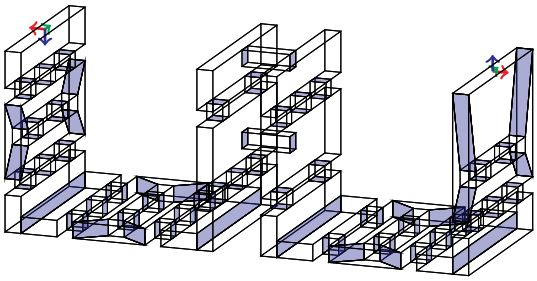
(f) Instance 6.



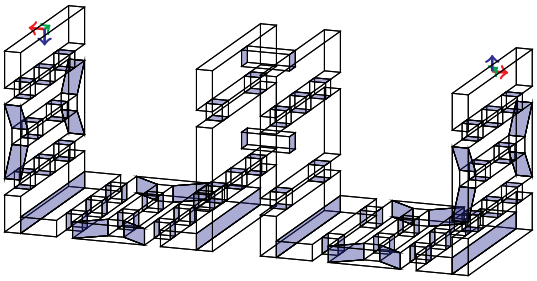
(g) Instance 7.



(h) Instance 8.

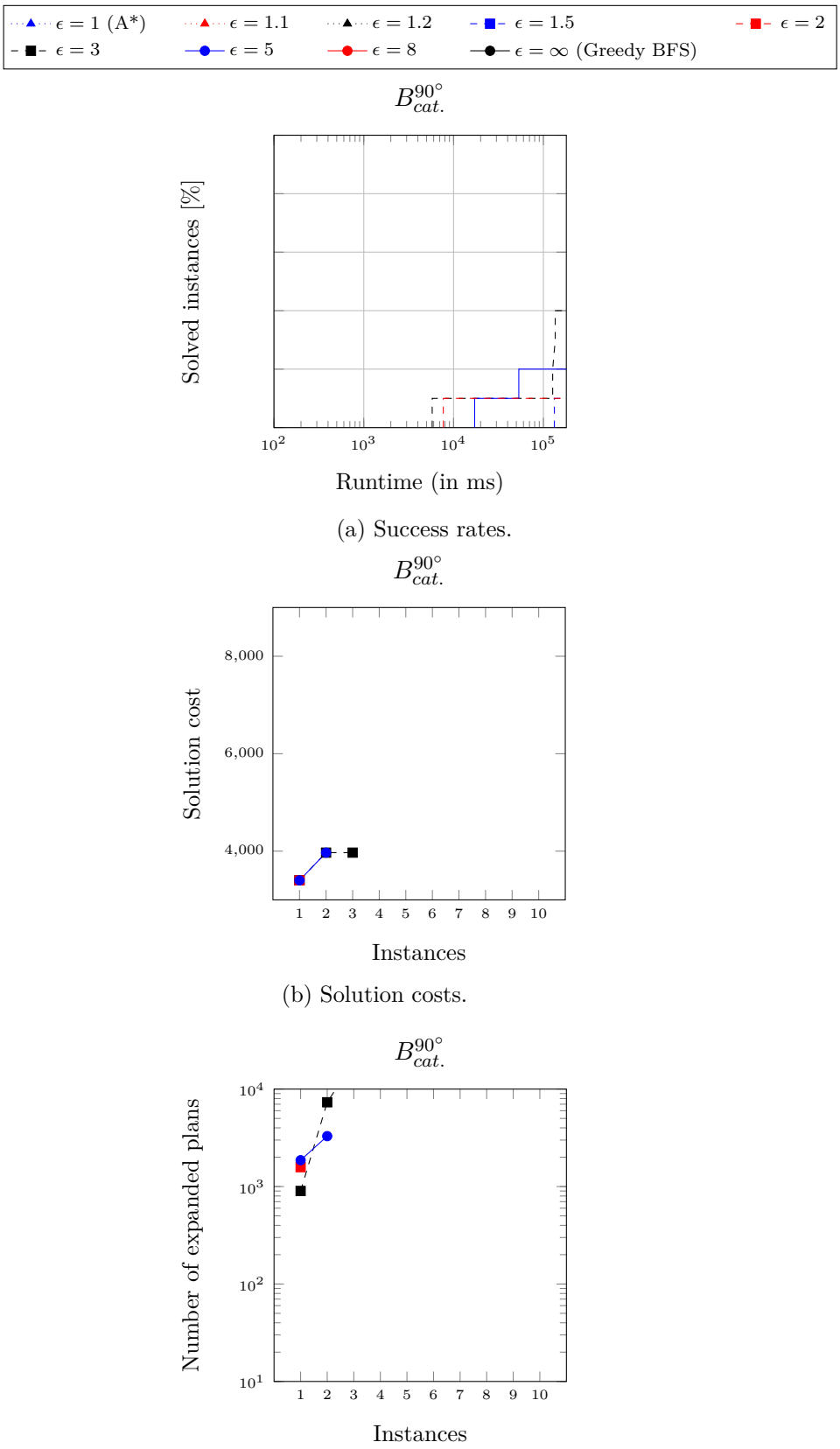


(i) Instance 9.



(j) Instance 10.

Figure 10.13 – Instances of the CWRP for the ISA approach.

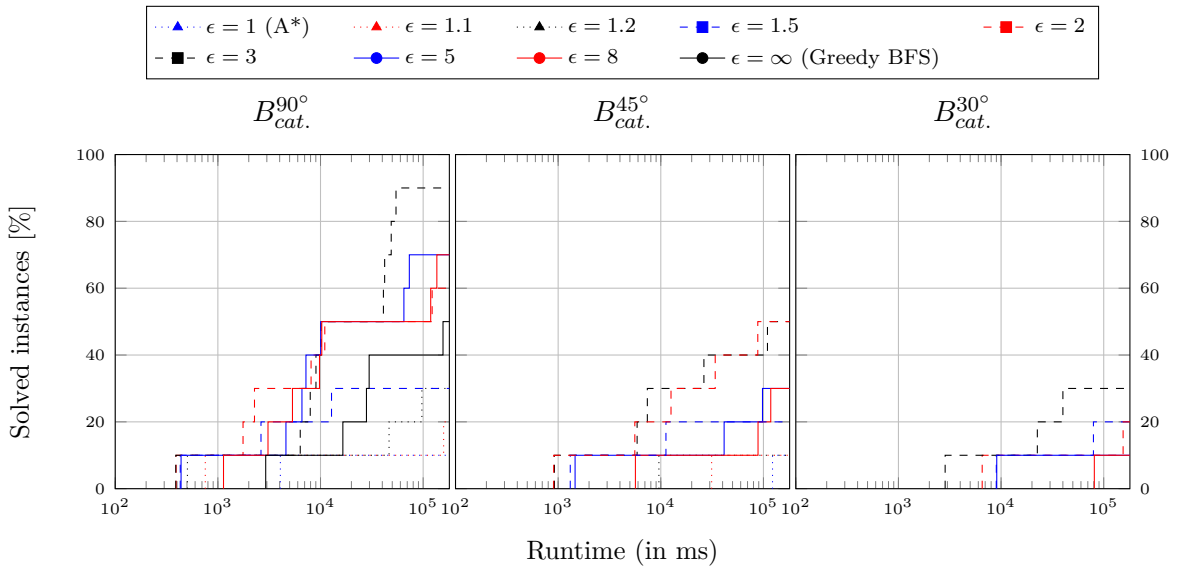


(a) Success rates.

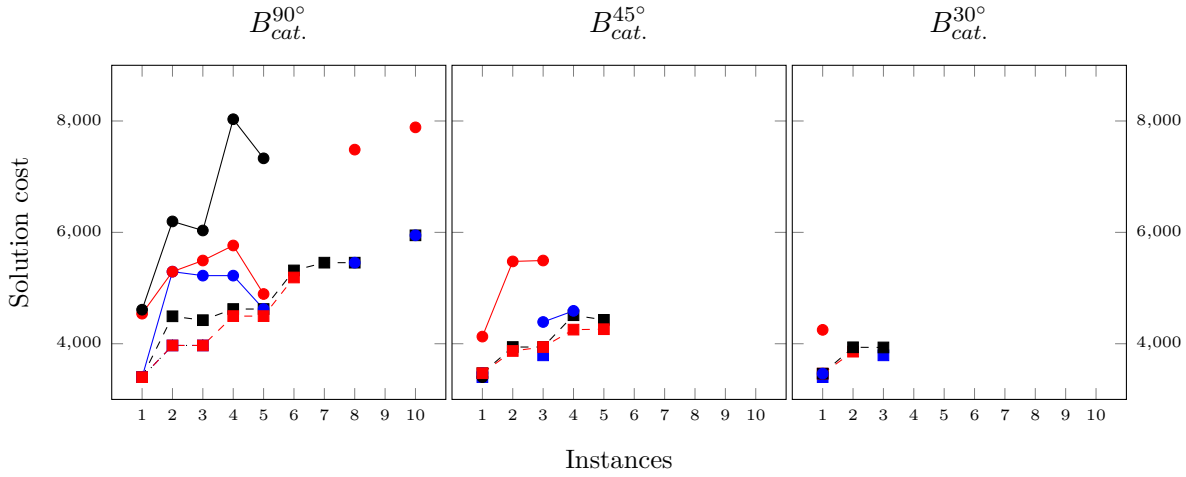
(b) Solution costs.

(c) Number of expanded plans.

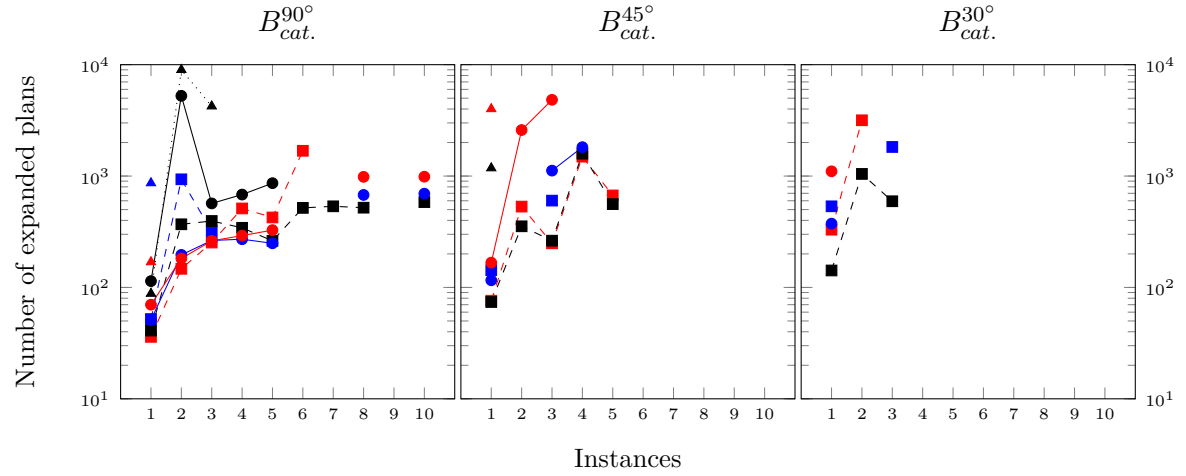
Figure 10.14 – Results with A*/WA* and the as the crow flies heuristic h_{ACF}.



(a) Success rates.

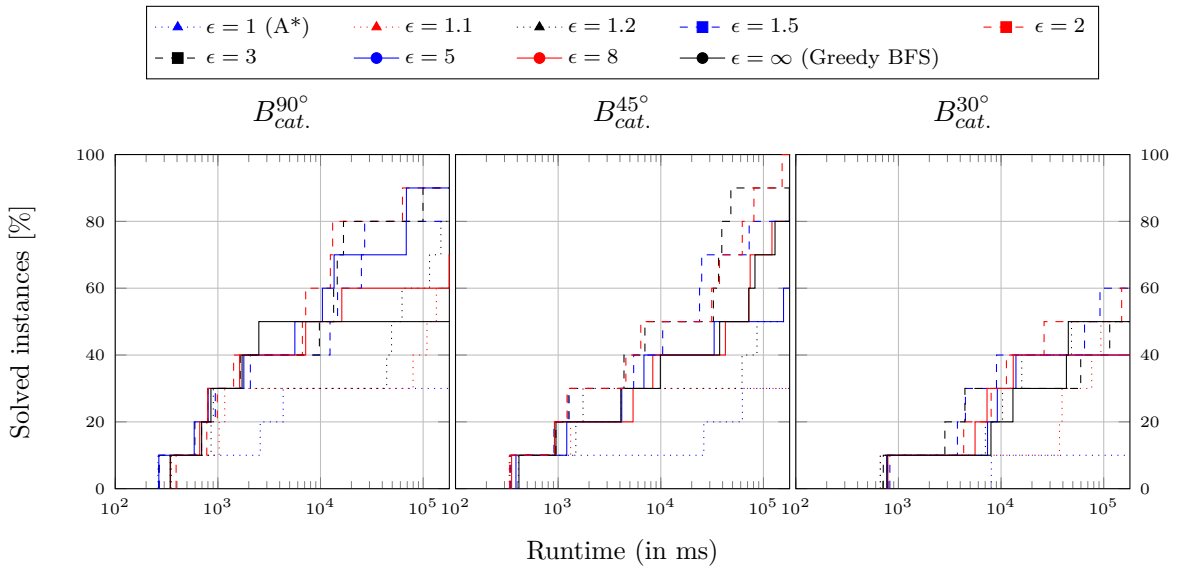


(b) Solution costs.

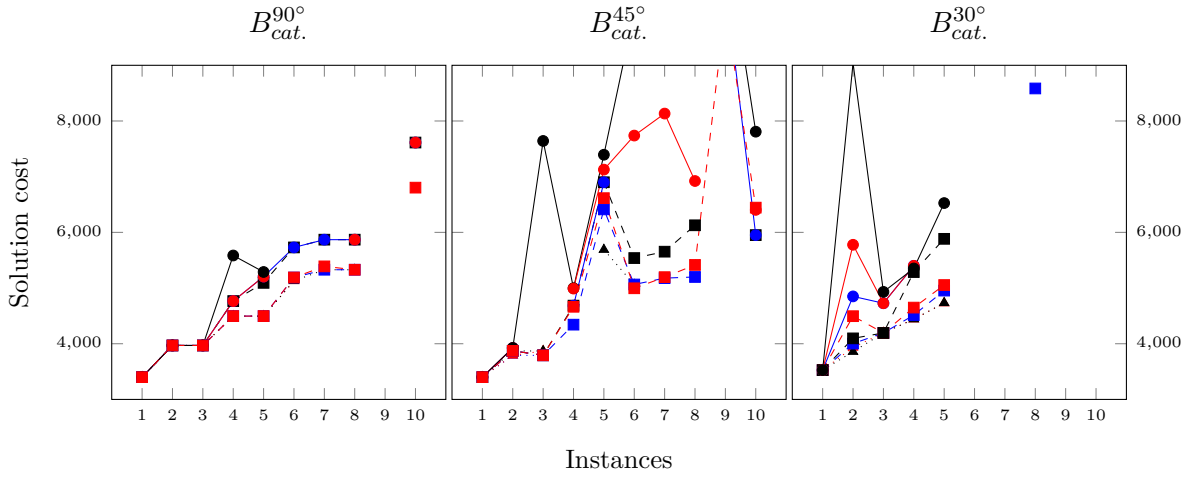


(c) Number of expanded plans.

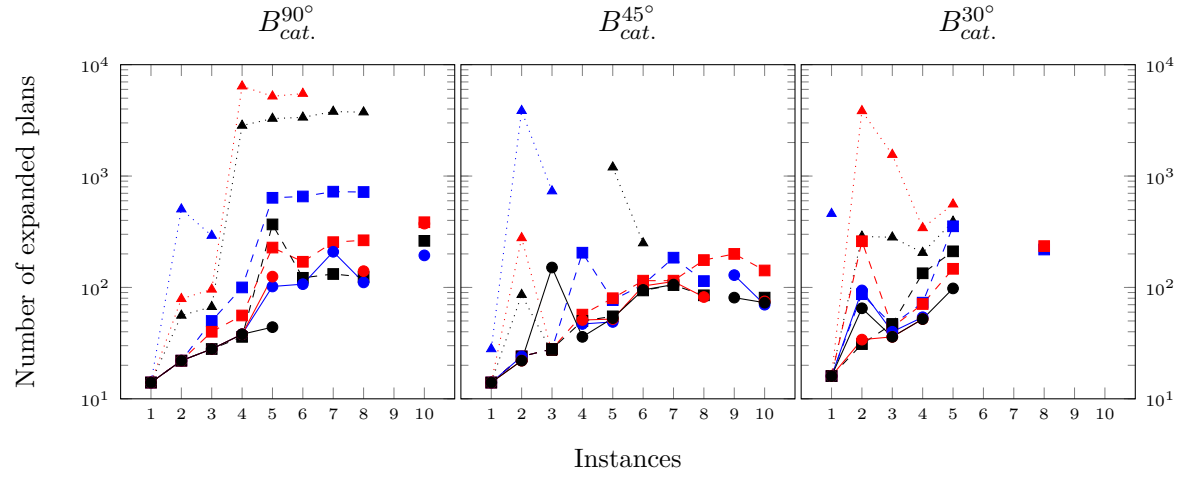
Figure 10.15 – Results with A*/WA* and the trail length heuristic h_{TL}.



(a) Success rates.

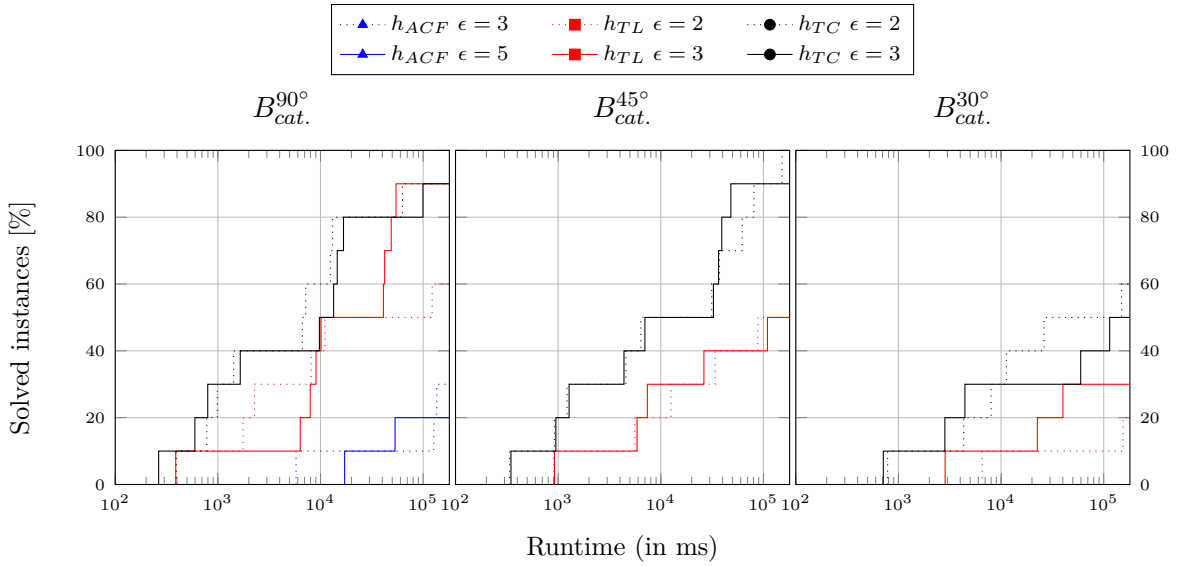


(b) Solution costs.

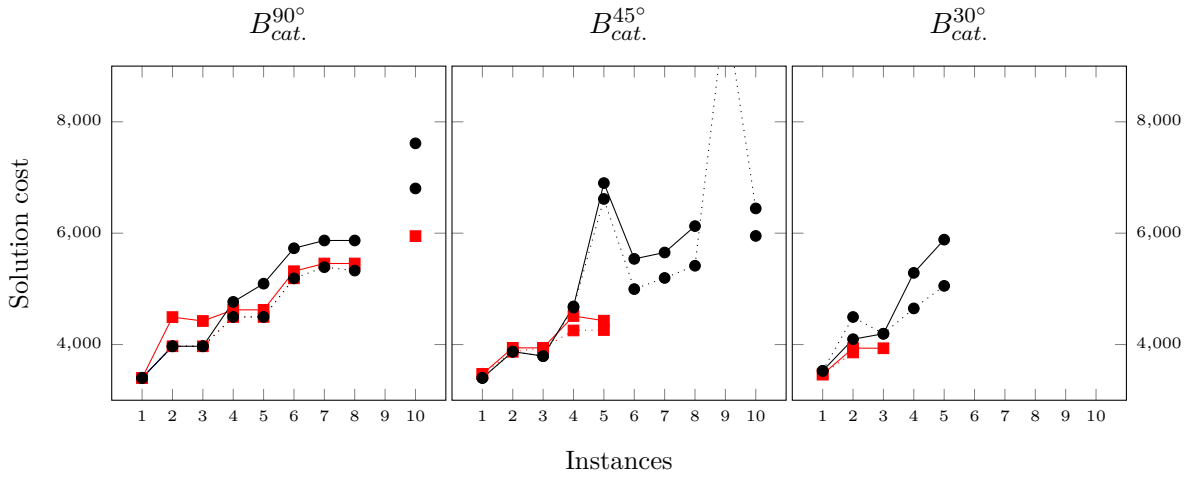


(c) Number of expanded plans.

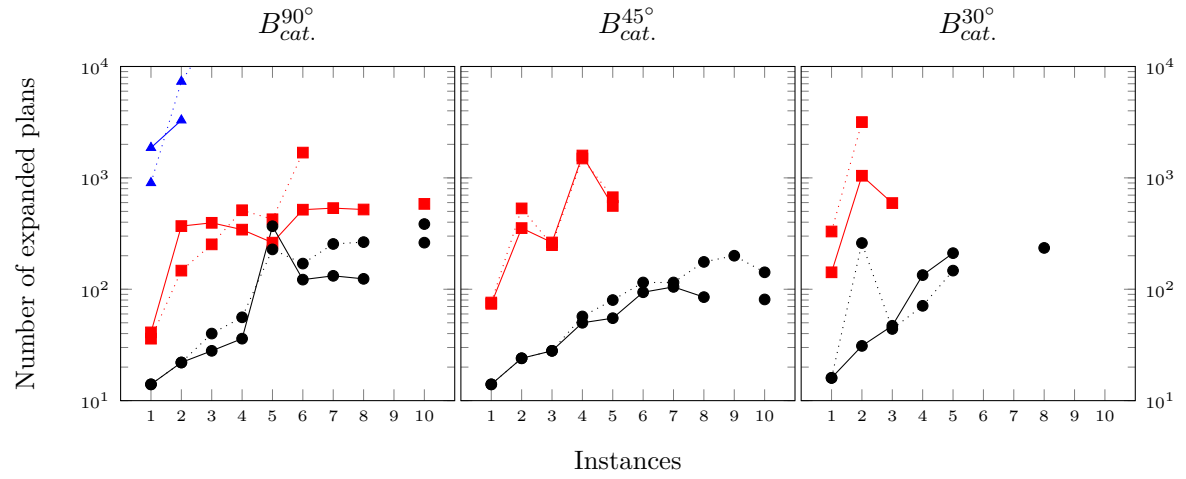
Figure 10.16 – Results with A*/WA* and the trail cost heuristic h_{TC}.



(a) Success rates.

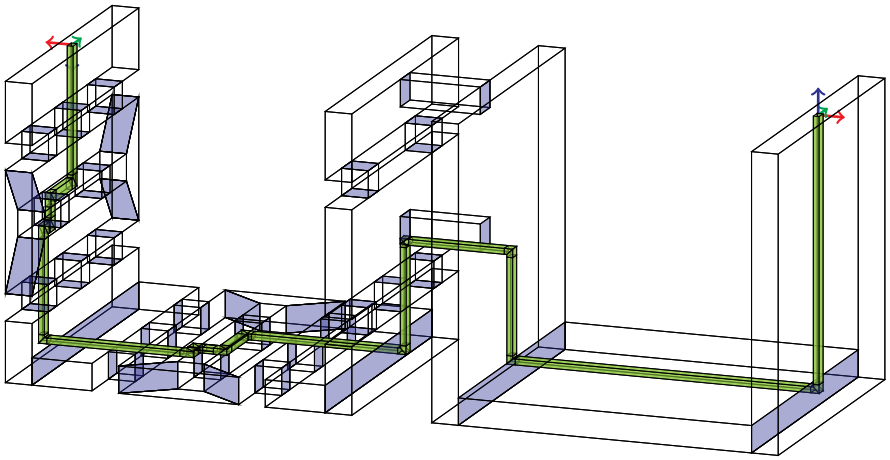


(b) Solution costs.

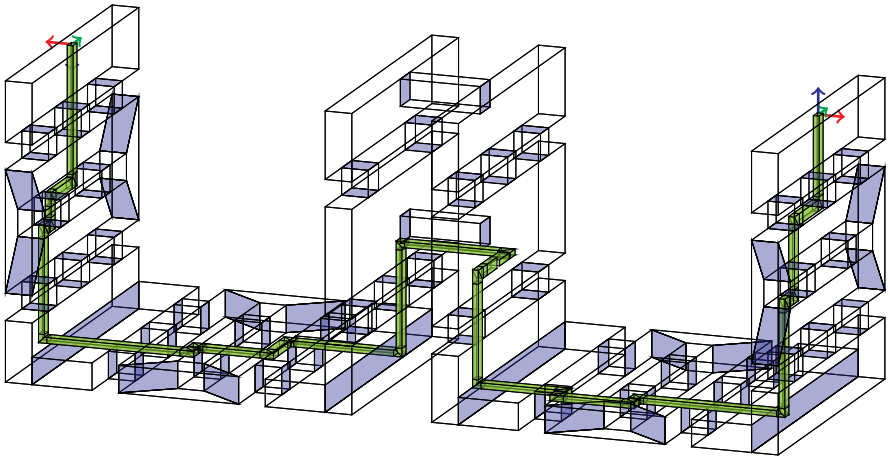


(c) Number of expanded plans.

Figure 10.17 – Best results with A*/WA*.



(a) Solution on instance 5.



(b) Solution on instance 10.

Figure 10.18 – Examples of routed waveguides with WA*.

10.3.3 Impact of the sampling radius

In a second series of experiments, the impact of the sampling radius ρ on the resolution performances is studied. First, Figure 10.19 on the following page presents the evolution of the trail space size with the sampling radius ρ for the test instances. The duration required to sample points on interfaces and to connect them in order to build the trail space is also reported.

Of course, the number of trail nodes raises when sampling radius ρ decreases. As a consequence, there are more nodes sampled on each interface and the evaluation of trail heuristics for a routing plan becomes more time-consuming since more trails must be compared to find the shortest one. In return, the estimation of the remaining distance to the destination is improved because the optimal path can be approximated with a smaller error. So better solutions are expected to be found when using a smaller sampling radius ρ. However, as shown on Figure 10.19, reducing the sampling radius on interfaces can significantly increase the trail space generation time. On the test instances, a ρ-value of 2 requires several minutes to build the trails which is already too excessive in comparison with the resolution time reached with $\rho = 10$ in Section 10.3.2.

Then, heuristic h_{TC} has been tested using WA* with $\epsilon = 2$ and ρ-values of 2, 5, 10, 20 and 50. The results are shown on Figure 10.20. Remind that the reported runtimes include the trail

Radius	2		5		10		20		50	
Instance	$\|\mathcal{M}\|$	Runtime (ms)	$\|\mathcal{M}\|$	Runtime (ms)	$\|\mathcal{M}\|$	Runtime (ms)	$\|\mathcal{M}\|$	Runtime (ms)	$\|\mathcal{M}\|$	Runtime (ms)
1	27182	139261	4499	2418	1200	125	335	16	64	1
2	40359	224952	6719	3541	1797	203	505	15	97	1
3	42793	180446	7156	2808	1924	171	541	16	100	1
4	53043	257339	8889	4025	2365	249	668	16	130	1
5	57944	251370	9727	3775	2619	219	741	15	135	1
6	65746	263306	11046	4149	2965	234	843	32	154	1
7	75974	374992	12759	5336	3422	327	974	31	175	1
8	78442	354586	13204	5709	3554	297	1015	47	173	1
9	88736	473274	14933	6366	4022	390	1149	31	206	1
10	88225	447279	14853	6521	4010	374	1147	47	199	1

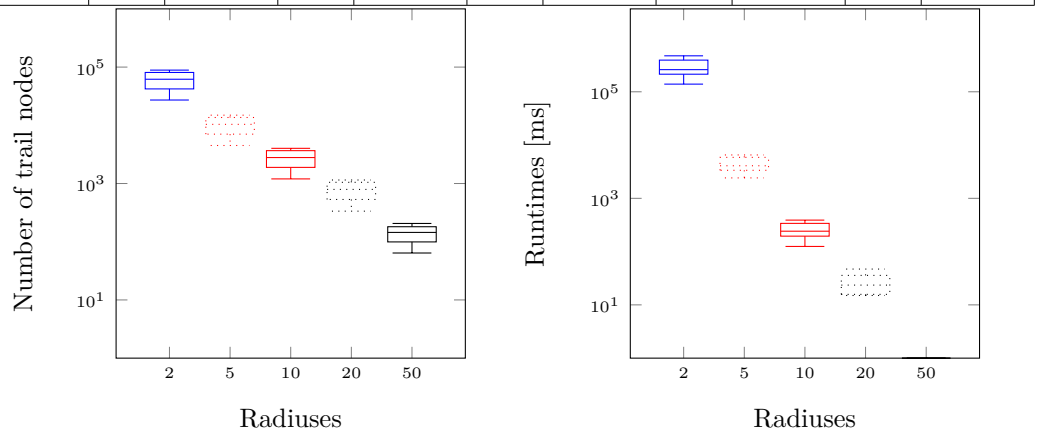
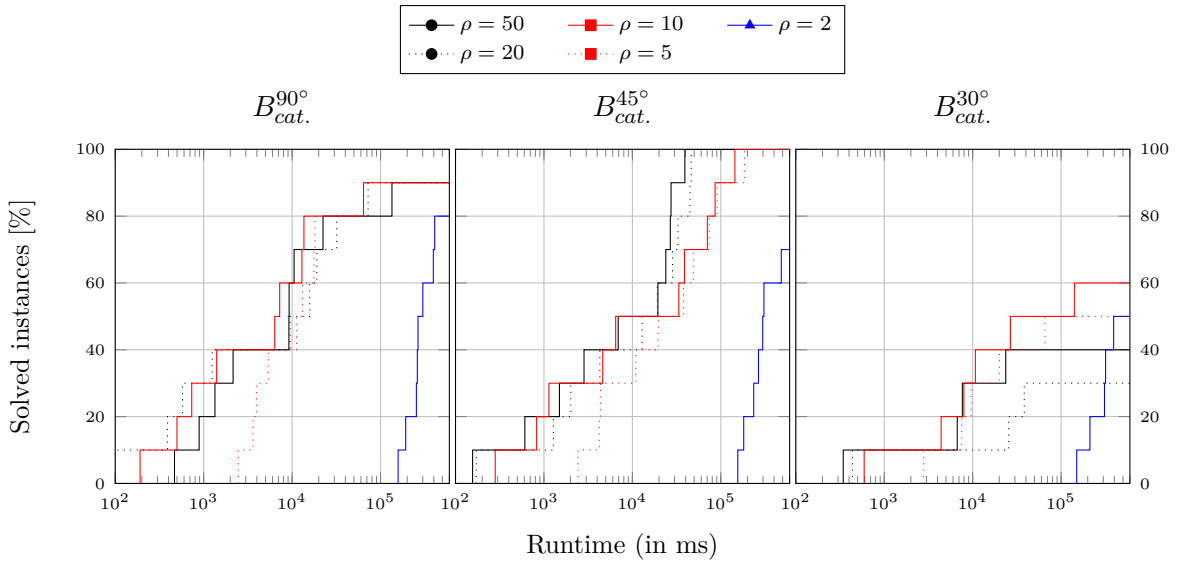


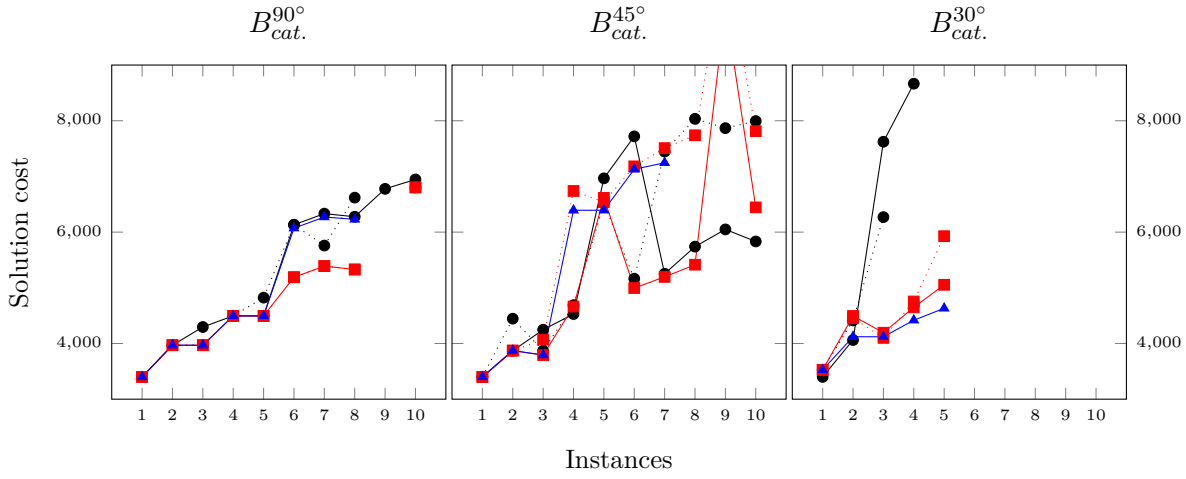
Figure 10.19 – Generation of the trail space.

space generation and the resolution for the trail heuristics. For convenience reasons, resolution time is limited to 3 minutes.

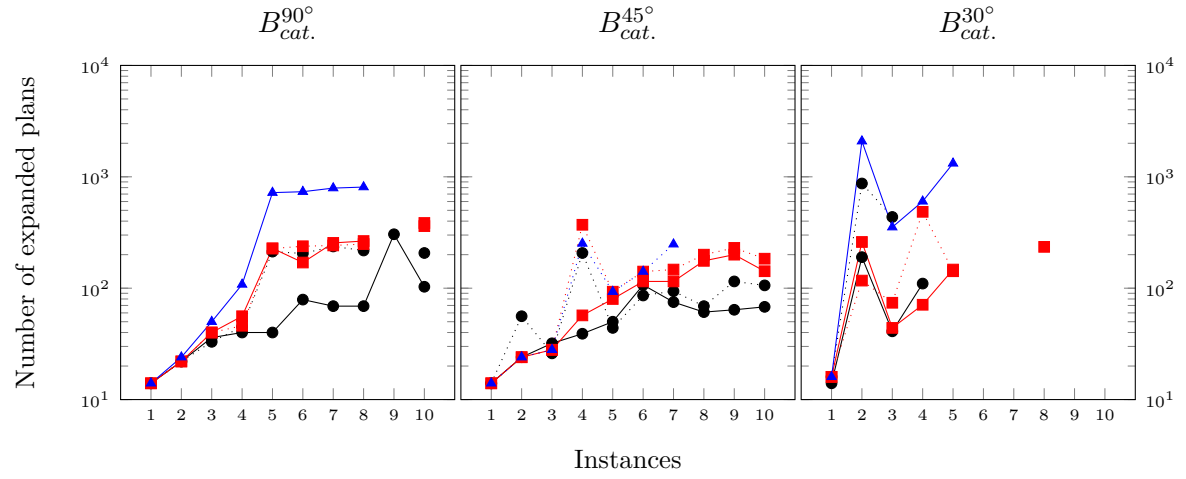
It appears that using a smaller sampling radius ρ seems to increase the number of visited routing plans and consequently the resolution time, as illustrated on Figure 10.20c. This can be explained by the more accurate estimation of the distance to the destination which tends to reduce the evaluation difference between two nearly similar plans. At the same time, reducing the sampling radius can improve the solution quality, as it can be verified on Figure 10.20b with bend catalogues $B^{90^\circ}_{cat.}$ and $B^{30^\circ}_{cat.}$. However, there is no guarantee that the solution found is better than when using a larger ρ-value (see the results with catalogue $B^{45^\circ}_{cat.}$). Actually, the effect of the high heuristic weight $\epsilon = 2$ that implies a more greedy exploration of the search space than pure A* makes the solution quality more random and mitigates the impact of the sampling radius. Finally, on the test instances, using WA* with $\epsilon = 2$ and $\rho = 10$ is the best trade-off between the solution quality and the whole resolution time including the trail space generation and best-first search.



(a) Success rates.



(b) Solution costs.



(c) Number of expanded plans.

Figure 10.20 – Results with different ρ-values.

Chapter 11

Conclusion on the CWRP

11.1 Contributions

In Part III, the Constrained Waveguide Routing Problem (CWRP), which extends the Free Waveguide Routing Problem (FWRP) with spatial constraints like the presence of obstacles, is addressed. To deal with these spatial constraints, a methodology based on triangulation is presented to decompose the available space into non-regular convex cells that avoid obstacles. Then, the Mixed Integer Linear Programming (MILP) and Search Problem (SP) formulations proposed for the FWRP are adapted to consider the sequence of cells crossed by the waveguide. Unlike the classical Pipe Routing techniques, these cells are only used to apply local constraints on the neutral fibre and the waveguide is still routed in a continuous domain using Linear Programming (LP). Solutions obtained with both formulations satisfy all waveguide design rules and avoid the obstacles of the routing environment. The SP formulation has been published at the CPAIOR 2021 [90] and ROADEF 2021 [91] conferences.

Again, solving the CWRP using the MILP formulation is shown to be highly time-consuming, even on very small instances which are not representative of the complexity of industrial cases. Indeed, when using a real bend catalogue, several hours may be necessary to find a first solution and optimality cannot be reached in a reasonable time. So the MILP approach tested is not a good candidate for an intensive usage during design phases.

In comparison, the resolution of the SP formulation using Informed Search Algorithms (ISAs) can provide good solutions within a few minutes on more realistic instances. Since the naive heuristic which consists in ignoring space constraints and using the as the crow flies distance to the destination performs very poorly, more sophisticated heuristics are introduced. They are based on a graph of relaxed routes, called trails, which connect each cell to the destination while staying inside the routing space. Thus, with the trail length heuristic, the remaining distance to the destination is more accurately estimated using the shortest trail for the evaluated routing plan. The trail cost heuristic goes further by estimating the cost of the best bend combination to use in order to follow the trail to the destination. A*, WA* and Greedy BFS have been experimented with these heuristic functions. It appears that the trail cost heuristic is the most robust and provides the best performances since it allows to solve realistic instances with an industrial bend catalogue within a few minutes. However, all these heuristics are not admissible, so there is no guarantee on the quality of the solutions provided by the ISAs even if, in practice, they are completely acceptable for waveguide designers.

As previously, the resolution approach of the CWRP based on ISAs has been implemented and integrated in an Airbus DS software suite and provided to the waveguide designers. If the proposed optimisation method finds pretty good solutions on simple cases, the resolution speed is unfortunately not fast enough to be efficiently used in a complete satellite environment.

11.2 Perspectives

To obtain better performances on the CWRP, several perspectives can be explored to improve the optimisation methods introduced in this thesis. First, in the SP formulation, the linear programs that evaluate feasibility and cost are solved from scratch for each generated routing plan. Warm-starting them using the solution found for the prior plan could significantly boost the evaluation of routing plans and reduce the resolution time with ISAs.

Furthermore, the evaluation functions might be more accurate by adding to the routing plan the definition of the next interface that should be crossed. The underlying idea is that the linear programs could exploit such a specification to generate a better intermediate end point.

Last, the resolution of the CWRP with ISAs suffers from the high branching factor implied by the bend catalogue size and the decisions to be taken in the routing space. A possible way to address this issue could be to focus the exploration inside the channel formed by the shortest trail between the origin and destination configurations. As a consequence, the possible decisions from a routing plan would be reduced to the addition of bends and the crossing of the next interface in the channel. Of course, the routing may be infeasible if the channel is too narrow and, in this case, another channel should be selected. A similar idea could be applied in the MILP formulation using a column generation technique based on a sub-problem which generates candidate routing channels. With this advanced LP method, the MILP formulation might become competitive.

Part IV

Conclusion

Chapter 12

General Conclusion

12.1 Contributions

In this thesis, several optimisation methods are proposed to automate the detailed routing of waveguides in the radio-frequency harness of a telecommunication satellite. The originality of these approaches, in comparison with classical pipe routing algorithms, is to deal with non-orthogonal bends defined by a catalogue and with unsymmetrical cross-sections. Indeed, these aspects are necessary to provide standardised solutions that satisfy all the constraints of waveguide routing in a satellite payload. Moreover, the introduced approaches use linear programming to route the waveguide in a 3D continuous domain, considering constraints which would be hard to express in a method based on a graph of candidate routes. Thus, the generated solutions are more realistic and can be used without modification by waveguide designers.

The waveguide routing problem is addressed in two steps. First, all spatial constraints and obstacles are ignored in a simplified version of the problem. This one can be solved using mixed integer linear programming, but this approach does not perform well enough to be applied in an industrial context. By contrast, a significantly more efficient resolution is proposed using informed search algorithms, like WA*, with an admissible heuristic that can ensure a bound on the solution quality. The integration of this last method in a design software noticeably reduced the duration of the waveguide routing phase for a real satellite.

In a second step, both introduced formulations are extended to deal with spatial constraints and obstacles. To do so, the available routing space is decomposed into convex cells which are used to generate local constraints on the waveguide neutral fibre. For this more complex problem, only the formulation using informed search algorithms is able to provide solutions for instances that are representative of industrial cases. However, the resolution of such instances requires several minutes which is not fast enough to make an intensive usage possible during the numerous iterations performed by designers.

12.2 Perspectives

Although the proposed approach to deal with obstacles can be improved, a more promising idea may consist in taking advantage of the good performances provided by the routing algorithm in free space to mimic designers' operations. Indeed, in practice, they decompose the route of a waveguide using waypoints in order to manually avoid conflicts with other components before applying the routing algorithm. So an interesting way to address the waveguide routing problem can be to generate an initial route without considering conflicts and then to modify it by adding, deleting or moving waypoints with a simulated annealing or genetic algorithm in order to fix these conflicts. Each time a waypoint is modified, the proposed routing algorithm in free space

can be used to compute the new waveguide detailed route. Then a fitness function may evaluate the quality of a solution depending on the number of conflicts between the waveguide and other components.

Moreover, the main asset of this method based on waypoints is that it can be easily extended to address the multiple waveguide routing problem. This one consists in routing several waveguides in a routing space which contains obstacles while avoiding conflicts between waveguides. Generally, a safety distance must be ensured between waveguides but it is also interesting to group them locally in order to share common brackets and reduce the cost of the RF-harness. The classical approach which consists in routing waveguides one at a time considering the previously routed ones as obstacles leads to suboptimal solutions that depend on the order in which waveguides are routed. Instead of this sequential approach, initial conflicting routes can be generated with the routing algorithm in free space and then the previous conflict resolution method based on waypoints can be used to manipulate the routes of the different waveguides. In this case, the fitness function must also take conflicts between waveguides into account. Furthermore, operations that merge waypoints from different waveguides may be proposed to model common sections shared between several routes.

Last, whatever the routing approach used, the quality of waveguide routes always depends on the available routing space that results from the position of other components on the satellite walls. Ideally, the component layout and waveguide routing problems should be solved simultaneously in order to improve the quality of the RF-harness. For instance, component moving operations can be introduced in the proposed conflict resolution method based on waypoints. However, the joint problem is hard to solve because of the huge research space induced by the number of waveguides and components in an industrial context. Furthermore, many other physical constraints involved in the component layout of a satellite payload, like thermal or electromagnetic constraints which can restrict the position of components to predefined areas or require minimal or maximal distances between particular kinds of components. It results that a multi-disciplinary optimization approach would be needed to address the complete payload design of a telecommunication satellite.

Part V

Appendix

Glossary

Abbreviations and acronyms

3D-ESPP 3D Euclidean Shortest Path Problem.

A* A* Search.

ACO Ant Colony Optimisation.

AI Artificial Intelligence.

Airbus DS Airbus Defence and Space.

AIT Assembly Integration and Test.

ARA* Anytime Repairing A* Search.

ATA* Anytime A* Search.

BDS Bi-Directional Search.

BF-BS Best-First Beam Search.

BFS Best-First Search.

BrF-BS Breadth-First Beam Search.

BrFS Breadth-First Search.

BS Beam Search.

BSA Blind Search Algorithm.

BSS Beam-Stack Search.

BULB Beam search Using Limited discrepancy Backtracking.

CABS Complete Anytime Beam Search.

CAD Computer-Aided Design.

CDA Cell Decomposition Approach.

CIFRE Industrial Agreement of Training through Research or "Convention Industrielle de Formation par la REcherche" in French.

CP Constraint Programming.

CWRP Constrained Waveguide Routing Problem.

D/C Down Converter.

DF-BS Depth-First Beam Search.

DFBnB Depth-First Branch-and-Bound Search.

DFID Depth-First Iterative Deepening Search.

DFS Depth-First Search.

EA Escape Algorithm.

EHC Enforced Hill-Climbing.

ESPP Euclidean Shortest Path Problem.

FWRP Free Waveguide Routing Problem.

GA Genetic Algorithm.

Greedy Greedy Search.

Greedy BFS Greedy Best-First Search.

HC Hill-Climbing.

HS Heuristic Search.

IDA* Iterative Deepening A* Search.

IMUX Input Multiplexer.

ISA Informed Search Algorithm.

KBE Knowledge Based Engineering.

LNA Low Noise Amplifier.

LP Linear Programming.

LSA Line Search Algorithm.

LSS-LRTA* Local Search Space Learning Real-Time A*.

MILP Mixed Integer Linear Programming.

MP Mathematical Programming.

MRA Maze Routing Algorithm.

MSC-kwA* Multi-State Commitment k-Weighted A* Search.

MWRP Multiple Waveguide Routing Problem.

OMUX Output Multiplexer.

ONERA French Aerospace Lab or "Office National d'Études et de Recherches Aérospatiales" in French.

PM Parametric Model.

PR Pipe Routing.

PRP Pipe Routing Problem.

PSO Particle Swarm Optimisation.

RBFS Recursive Best-First Search.

RF-harness Radio-Frequency Harness.

RF-losses Radio-Frequency Losses.

RTS Real-Time Search.

SA Search Algorithm.

SAHC Steeple Ascent Hill-Climbing.

SHC Simple Hill-Climbing.

SkA Skeleton Approach.

SP Search Problem.

StHC Stochastic Hill-Climbing.

TWTA Travelling Wave Tube Amplifier.

UCS Uniform Cost Search.

USA Uninformed Search Algorithm.

WA* Weighted A* Search.

Window A* Window A* Search.

WR Waveguide Routing.

WRP Waveguide Routing Problem.

Geometrical notations

Spaces

Sign	*Domain*	*Description*
$\mathbb{R}^3$		Three-dimensional canonical Euclidean space
$\mathbb{R}^2$		Two-dimensional canonical Euclidean space

Vectors

Sign	*Domain*	*Description*
$\overrightarrow{e_x}$	$\mathbb{R}^3$	Reference X-axis vector
$\overrightarrow{e_y}$	$\mathbb{R}^3$	Reference Y-axis vector
$\overrightarrow{e_z}$	$\mathbb{R}^3$	Reference Z-axis vector

Orientations

Sign	*Domain*	*Description*
$\mathcal{O}$		Set of orientations of $\mathbb{R}^3$
$\overrightarrow{e_{o,x}}$	$\mathbb{R}^3$	$\overrightarrow{e_x}$ axis vector of orientation $o \in \mathcal{O}$
$\overrightarrow{e_{o,y}}$	$\mathbb{R}^3$	$\overrightarrow{e_y}$ axis vector of orientation $o \in \mathcal{O}$
$\overrightarrow{e_{o,z}}$	$\mathbb{R}^3$	$\overrightarrow{e_z}$ axis vector of orientation $o \in \mathcal{O}$
o_{ref}	$\mathcal{O}$	Reference orientation $(\overrightarrow{e_x}, \overrightarrow{e_y}, \overrightarrow{e_z})$
M_o		Rotation matrix expressing orientation $o \in \mathcal{O}$ in reference orientation o_{ref}

Configurations

Sign	*Domain*	*Description*
Θ		Set of configurations of $\mathbb{R}^3$
P_θ	$\mathbb{R}^3$	Point of configuration $\theta \in \Theta$
$\overrightarrow{e_{\theta,x}}$	$\mathbb{R}^3$	$\overrightarrow{e_x}$ axis vector of configuration $\theta \in \Theta$
$\overrightarrow{e_{\theta,y}}$	$\mathbb{R}^3$	$\overrightarrow{e_y}$ axis vector of configuration $\theta \in \Theta$
$\overrightarrow{e_{\theta,z}}$	$\mathbb{R}^3$	$\overrightarrow{e_z}$ axis vector of configuration $\theta \in \Theta$
θ_{ref}	Θ	Reference configuration $(O, \overrightarrow{e_x}, \overrightarrow{e_y}, \overrightarrow{e_z})$

Transformations

Sign	*Domain*	*Description*
Id	$\Theta \to \Theta$	Identity

tr_L	$\Theta \to \Theta$	Translation of length $L \in \mathbb{R}^+$
rot_M	$\Theta \to \Theta$	Rotation of matrix M

Polyhedrons

Sign	*Domain*	*Description*
$\mathcal{P}$	$\mathcal{P} \subseteq \mathbb{R}^3$	Polyhedron
$\mathcal{Q}_\mathcal{P}$		Set of canonical equations of polyhedron $\mathcal{P} \subset \mathbb{R}^3$
a_q	$\mathbb{R}^+$	Coefficient a for equation $q \in \mathcal{Q}_\mathcal{P}$ of polyhedron $\mathcal{P} \subset \mathbb{R}^3$
b_q	$\mathbb{R}^+$	Coefficient b for equation $q \in \mathcal{Q}_\mathcal{P}$ of polyhedron $\mathcal{P} \subset \mathbb{R}^3$
c_q	$\mathbb{R}^+$	Coefficient c for equation $q \in \mathcal{Q}_\mathcal{P}$ of polyhedron $\mathcal{P} \subset \mathbb{R}^3$
d_q	$\mathbb{R}^+$	Coefficient d for equation $q \in \mathcal{Q}_\mathcal{P}$ of polyhedron $\mathcal{P} \subset \mathbb{R}^3$

Modelling of a waveguide

Straight sections

Sign	*Domain*	*Description*
$\mathcal{U}$		Set of straight sections in $\mathbb{R}^3$
L_u	$\mathbb{R}^+$	Length of straight section $u \in \mathcal{U}$
$\mathcal{F}_{u,\theta}$	$\mathbb{R}^3$	Neutral fibre of straight section $u \in \mathcal{U}$ applied from configuration $\theta \in \Theta$
$o_{u,\theta}$	$\mathcal{F}_{u,\theta} \to \mathcal{O}$	Orientation function of straight section $u \in \mathcal{U}$ applied from configuration $\theta \in \Theta$

Bends

Sign	*Domain*	*Description*
$\mathcal{B}$		Set of bends in $\mathbb{R}^3$
L_b	$\mathbb{R}^+$	Half-length of bend $b \in \mathcal{B}$
M_b		Rotation matrix of bend $b \in \mathcal{B}$
$\mathcal{F}_{b,\theta}$	$\mathbb{R}^3$	Neutral fibre of bend $b \in \mathcal{B}$ applied from configuration $\theta \in \Theta$
$o_{b,\theta}$	$\Theta \to \Theta$	Orientation function of bend $b \in \mathcal{B}$ applied from configuration $\theta \in \Theta$
α_b	$\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$	Angle of bend $b \in \mathcal{B}$
ρ_b	$\mathbb{R}^+$	Radius of bend $b \in \mathcal{B}$
$b_{neut.}$	$\mathcal{B}$	Neutral bend

Orientation changes

Sign	*Domain*	*Description*
$\mathcal{R}$		Set of orientation changes
o_r^-	$\mathcal{O}$	Origin orientation of orientation change $r \in \mathcal{R}$
o_r^+	$\mathcal{O}$	Destination orientation of orientation change $r \in \mathcal{R}$
b_r	$\mathcal{B}$	Bend of orientation change $r \in \mathcal{R}$

Waveguides

Sign	*Domain*	*Description*
Π		Set of waveguides in $\mathbb{R}^3$
$\mathcal{S}_\pi$	$\mathbb{R}^2$	Cross-section of waveguide $\pi \in \Pi$

$\mathcal{F}_\pi$	$\mathbb{R}^3$	Neutral fibre of waveguide $\pi \in \Pi$
o_π	$\mathcal{F}_\pi \to \mathcal{O}$	Orientation function on neutral fibre $\mathcal{F}_\pi$ of waveguide $\pi \in \Pi$
θ_π	$\mathcal{F}_\pi \to \Theta$	Configuration function on neutral fibre $\mathcal{F}_\pi$ of waveguide $\pi \in \Pi$
N_π	$\mathbb{N}^*$	Number of segments of neutral fibre $\mathcal{F}_\pi$ of waveguide $\pi \in \Pi$
$P_{\pi,k}$	$\mathbb{R}^3$	k^{th} point on neutral fibre $\mathcal{F}_\pi$ of waveguide $\pi \in \Pi$, for $k \in [\![1, N_\pi + 1]\!]$
$o_{\pi,k}$	$\mathcal{O}$	Orientation of the k^{th} segment on neutral fibre $\mathcal{F}_\pi$ of waveguide $\pi \in \Pi$, for $k \in [\![1, N_\pi]\!]$
$\ell_{\pi,k}$	$\mathbb{R}^+$	Length of the k^{th} segment on neutral fibre $\mathcal{F}_\pi$ of waveguide $\pi \in \Pi$, for $k \in [\![1, N_\pi]\!]$
$u_{\pi,k}$	$\mathcal{U}$	k^{th} straight section of waveguide $\pi \in \Pi$, for $k \in [\![1, N_\pi]\!]$
$b_{\pi,k}$	$\mathcal{B}$	k^{th} bend of waveguide $\pi \in \Pi$, for $k \in [\![1, N_\pi - 1]\!]$
$r_{\pi,k}$	$\mathcal{R}$	k^{th} orientation change of waveguide $\pi \in \Pi$, for $k \in [\![1, N_\pi - 1]\!]$

Problem inputs

Free Waveguide Routing Problem

Sign	*Domain*	*Description*
$\mathcal{P}^{ori.}$	$\mathbb{R}^3$	Origin polyhedron
$o^{ori.}$	$\mathcal{O}$	Origin orientation
$\mathcal{P}^{dest.}$	$\mathbb{R}^3$	Destination polyhedron
$\mathcal{O}^{dest.}$	$\mathcal{O}$	Set of possible destination orientations
$B_{cat.}$	$\mathcal{B}$	Catalogue of available bends
N_S	$\mathbb{N}^*$	Maximum number of segments for the neutral fibre
L^{min}	$\mathbb{R}^+$	Minimum length of straight sections
$\mathcal{O}^{int.}$	$\mathcal{O}$	Set of intrinsically attachable orientations
μ	$\mathbb{R}^+$	Linear cost
γ_b	$\mathbb{R}^+$	Cost of bend $b \in B_{cat.}$
$c^{ori.}$	$\mathcal{C}$	Origin traversable cell
$\mathcal{C}^{dest.}$	$\mathcal{C}$	Set of destination traversable cells
N_C	$\mathbb{N}^*$	Number of traversable cells

Reachable orientations

Sign	*Domain*	*Description*
$\mathcal{O}_\infty$	$\mathcal{O}$	Set of reachable orientations
$\mathcal{R}_\infty$	$\mathcal{R}$	Set of reachable orientation changes
$\mathcal{R}_\infty^{inc.}(o)$	$\mathcal{R}$	Set of reachable orientation changes incoming into reachable orientation $o \in \mathcal{O}_\infty$
$\mathcal{R}_\infty^{out.}(o)$	$\mathcal{R}$	Set of reachable orientation changes outgoing from reachable orientation $o \in \mathcal{O}_\infty$
$G(\mathcal{O}_\infty, \mathcal{R}_\infty)$		Kernel of reachable orientations

Candidate orientations

Sign	*Domain*	*Description*
$\mathcal{O}_k$	$\mathcal{O}_\infty$	Set of k-candidate orientations, for $k \in \mathbb{N}^*$
$\mathcal{R}_k$	$\mathcal{R}_\infty$	Set of k-candidate orientation changes, for $k \in \mathbb{N}^*$
$\mathcal{R}_k^{inc.}(o)$	$\mathcal{R}_{k-1}$	Set of candidate orientation changes incoming into the candidate orientation $o \in \mathcal{O}_k$, for $k \in [\![1, N_S]\!]$
$\mathcal{R}_k^{out.}(o)$	$\mathcal{R}_k$	Set of candidate orientation changes outgoing from the candidate orientation $o \in \mathcal{O}_k$, for $k \in [\![1, N_S]\!]$

$G\left(\mathcal{O}_1^{N_S}, \mathcal{R}_1^{N_S-1}\right)$	Space of candidate orientations for the FWRP
$G\left(\mathcal{O}_1^{K+1}, \mathcal{R}_1^{K}\right)$	Space of candidate orientations for the CWRP

Routable space

Sign	*Domain*	*Description*
$\mathcal{C}$	$\mathbb{R}^3$	Set of traversable cells
$\mathcal{P}_c$	$\mathbb{R}^3$	Polyhedron of traversable cell $c \in \mathcal{C}$
$\mathcal{I}$	$\mathbb{R}^3$	Set of crossable interfaces
$\mathcal{P}_i$	$\mathbb{R}^3$	Polyhedron of crossable interface $i \in \mathcal{I}$
$\overrightarrow{n_i}$	$\mathbb{R}^3$	Normal of crossable interface $i \in \mathcal{I}$
$reverse\,(i)$	$\mathbb{R}^3$	Reverse interface of crossable interface $i \in \mathcal{I}$
c_i^-	$\mathcal{C}$	Origin traversable cell of crossable interface $i \in \mathcal{I}$
c_i^+	$\mathcal{C}$	Destination traversable cell of crossable interface $i \in \mathcal{I}$
$i_{neut.}^c$	$\mathcal{I}$	Neutral crossable interface staying into traversable cell $c \in \mathcal{C}$ with $\mathcal{P}_{i_{neut.}^c} = \mathcal{P}_c$
$\mathcal{I}_c^{inc.}$	$\mathcal{I}$	Set of crossable interfaces incoming into traversable cell $c \in \mathcal{C}$
$\mathcal{I}_c^{out.}$	$\mathcal{I}$	Set of crossable interfaces outgoing from traversable cell $c \in \mathcal{C}$
$G\,(\mathcal{C}, \mathcal{I})$	$\mathbb{R}^3$	Routable space

Candidate cells

Sign	*Domain*	*Description*
$\mathcal{C}_k$	$\mathcal{C}$	Set of k-candidate traversable cells, for $k \in \mathbb{N}^*$
$\mathcal{I}_k$	$\mathcal{I}$	Set of k-candidate crossable interfaces, for $k \in \mathbb{N}^*$
$\mathcal{I}_k^{inc.}\,(o)$	$\mathcal{I}_{k-1}$	Set of candidate crossable interfaces incoming into the candidate traversable cell $o \in \mathcal{O}_k$, for $k \in [\![1, N_S]\!]$
$\mathcal{I}_k^{out.}\,(o)$	$\mathcal{I}_k$	Set of candidate crossable interfaces outgoing from the candidate traversable cell $o \in \mathcal{O}_k$, for $k \in [\![1, N_S]\!]$
$G\left(\mathcal{C}_1^{K+1}, \mathcal{I}_1^{K}\right)$		Space of candidate traversable cells

Algorithms

Mixed Integer Linear Programming

Sign	*Domain*	*Description*
$x_{k,r}$	$\{0,1\}$	Boolean variable equal to 1 if orientation change $r \in \mathcal{R}_\infty k$ is applied at the end point of the k^{th} segment on neutral fibre $\mathcal{F}_\pi$, 0 otherwise, for $k \in [\![1, K-1]\!]$
$y_{k,i}$	$\{0,1\}$	Boolean variable equal to 1 if the interface $i \in \mathcal{I}_k$ is crossed by the k^{th} segment on neutral fibre $\mathcal{F}_\pi$, 0 otherwise, for $k \in [\![1, K]\!]$
$z_{k,k'}$	$\{0,1\}$	Boolean variable equal to 1 if there is an straight section between the k^{th} and k'^{th} vertices of neutral fibre $\mathcal{F}_\pi$, 0 otherwise, for $k \in [\![1, K+1]\!]$ and $k' \in [\![k+1, K+2]\!]$
ℓ_k	$\mathbb{R}^+$	Length of the k^{th} segment on neutral fibre $\mathcal{F}_\pi$, for $k \in [\![1, N_S]\!]$
p_k	$\mathbb{R}^3$	Position of the k^{th} vertex on neutral fibre $\mathcal{F}_\pi$, for $k \in [\![1, K+1]\!]$
$M_{succ.}$	$\mathbb{R}^+$	Big-M value in the succession constraints of the MILP formulations
M_q	$\mathbb{R}^+$	Big-M value for equation $q \in \mathcal{Q}_\mathcal{P}$ of polyhedron $\mathcal{P}$ in the space constraints of the MILP formulations
$M_{len.}$	$\mathbb{R}^+$	Big-M value in the minimal length constraints of the MILP formulations
L_{UB}	$\mathbb{R}^+$	Upper bound of segment lengths on neutral fibre $\mathcal{F}_\pi$

Informed Search Algorithms

Sign	*Domain*	*Description*
$\mathcal{S}$		Set of routing plans
N_s	$\mathbb{N}^*$	Number of segments of routing plan $s \in \mathcal{S}$
$\mathcal{F}_s$	$\mathbb{R}^3$	Neutral fibre of routing plan $s \in \mathcal{S}$
p_k^s	$\mathbb{R}^3$	k^{th} point on neutral fibre $\mathcal{F}_s$ of routing plan $s \in \mathcal{S}$, for $k \in [\![1, N_s+1]\!]$
ℓ_k^s	$\mathbb{R}^+$	Length of the k^{th} segment on neutral fibre $\mathcal{F}_s$ of routing plan $s \in \mathcal{S}$, for $k \in [\![1, N_s]\!]$
r_k^s	$\mathcal{R}$	Orientation change applied at the end point of the k^{th} segment on neutral fibre $\mathcal{F}_s$ of routing plan $s \in \mathcal{S}$, for $k \in [\![1, N_s-1]\!]$

o_k^s	$\mathcal{O}$	Orientation of the k^{th} segment on neutral fibre $\mathcal{F}_s$ of routing plan $s \in \mathcal{S}$, for $k \in [\![1, N_s]\!]$
b_k^s	$\mathcal{B}$	Bend applied at the end point of the k^{th} segment on neutral fibre $\mathcal{F}_s$ of routing plan $s \in \mathcal{S}$, for $k \in [\![1, N_s - 1]\!]$
c_k^s	$\mathcal{C}$	Traversable cell associated with the end point of the k^{th} segment on neutral fibre $\mathcal{F}_s$ of routing plan $s \in \mathcal{S}$, for $k \in [\![1, N_s + 1]\!]$
$\mathcal{I}_k^s$	$\mathcal{I}$	Sequence of crossable interfaces crossed by the k^{th} segment on neutral fibre $\mathcal{F}_s$ of routing plan $s \in \mathcal{S}$, for $k \in [\![1, N_s]\!]$
$q_{k,i}^s$	$\mathbb{R}^3$	Intersection between the k^{th} segment on neutral fibre $\mathcal{F}_s$ of routing plan $s \in \mathcal{S}$ and interface $i \in \mathcal{I}_k^s$ it has to cross, for $k \in [\![1, N_s]\!]$
$\alpha_{k,i}^s$	$\mathbb{R}^+$	Distance between the k^{th} point on neutral fibre $\mathcal{F}_s$ of routing plan $s \in \mathcal{S}$ and intersection $q_{k,i}^s$ with interface $i \in \mathcal{I}_k^s$ it has to cross, for $k \in [\![1, N_s]\!]$

Search Algorithms

Sign	*Domain*	*Description*
$\mathcal{S}$		Search space or set of states
$s_{ori.}$	$\mathcal{S}$	Initial state
$isGoal\,(s)$		Goal test which tells whether state $s \in \mathcal{S}$ is a target state
$\mathcal{A}$		Set of actions
$successors\,(s)$	$\mathcal{S} \times \mathcal{A}$	Set of possible action-successor pairs from state $s \in \mathcal{S}$
d	$\mathbb{N}$	Depth of the shallowest goal node
b	$\mathbb{N}$	Branching factor or mean number of successors for the states in $\mathcal{S}$

Trails

Sign	*Domain*	*Description*
$\mathcal{T}_s$		Set of trails for a routing plan $s \in \mathcal{S}$
N_t	$\mathbb{N}^*$	Number of nodes in trail $t \in \mathcal{T}_s$ for a routing plan $s \in \mathcal{S}$
m_k^t	$\mathcal{M}$	k^{th} node of trail $t \in \mathcal{T}_s$ for a routing plan $s \in \mathcal{S}$, for $k \in [\![1, N_t]\!]$
K_t	$\mathbb{N}^*$	Number of interfaces crossed by trail $t \in \mathcal{T}_s$ for a routing plan $s \in \mathcal{S}$

i_k^t	$\mathcal{I}$	k^{th} interface crossed by trail $t \in \mathcal{T}_s$ for a routing plan $s \in \mathcal{S}$, for $k \in [\![1, N_t]\!]$

Trail space

Sign	*Domain*	*Description*
ρ	$\mathbb{R}^+$	Sampling radius
$\mathcal{M}$		Set of trail nodes
$\mathcal{M}(i)$	$\mathcal{M}$	Set of trail nodes on interface $i \in \mathcal{I}$
$\mathcal{M}\left(\mathcal{P}^{dest.}\right)$	$\mathcal{M}$	Set of trail nodes in destination polyhedron $\mathcal{P}^{dest.}$
i_m	$\mathcal{I}$	Interface on which trail node $m \in \mathcal{M}$ has been sampled
L_m	$\mathbb{R}^+$	Length of the shortest trail in $G(\mathcal{M}, \mathcal{D})$ from trail node $m \in \mathcal{M}$ to destination polyhedron $\mathcal{P}^{dest.}$
$G(\mathcal{M}, \mathcal{D})$		Space of candidate trails

Experimentation data

Processor and memory

All experiments presented in this report have been performed using a single thread on an Intel® Core i5-6500 CPU 3.20 GHz processor with 23.4 GB of RAM.

Libraries

All approaches presented in this report have been implemented in Java. The implementation of MILP formulations uses the Google OR-Tools library (version 9.1.9490) [73] and the SCIP solver (version 7.0.1) [32, 33]. The resolution of the LP models has been performed using the Simplex solver from the Apache Commons Math library (version 3.6.1) [3]

Gauges

Table 5 lists the properties of the main waveguide gauges used in telecommunication satellites. Note that the width a and the height b presented in the following table correspond to external dimensions that include the walls of the waveguide.

Gauge	*a (in mm)*	*b (in mm)*	*Band*	*Frequency range (in Ghz)*
WR229	61.37	32.283	S	3.30–4.90
WR137	38.05	18.999	C	5.85–8.20
WR90	24.13	11.43	X	8.20–12.40
WR75	20.32	10.79	X-Ku	10–15
WR62	17.799	9.89	Ku	12.40–18
WR28	9.112	5.556	Ka	26.50–40

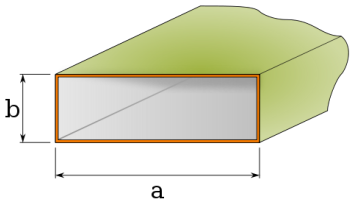


Table 5 – Common gauges in telecommunication satellites.

Instances of the FWRP

The origin and destination points of the FWRP instances used for experimentation are sampled in a parallelepiped whose dimensions are provided in Table 6.

Width (in mm)	*Depth (in mm)*	*Height (in mm)*
2750	2650	6250

Table 6 – Dimensions of the routable space.

Instances of the CWRP for the MILP approach

The CWRP instances of the experiments on the MILP approach are illustrated on Figure 9.2 on page 119 and use the routing space described on Table 7 and the origin and destination configurations described on Table 8. The rotation angles that define the orientations are expressed in the roll-pitch-yaw convention.

Cell	X_{min}	X_{max}	Y_{min}	Y_{max}	Z_{min}	Z_{max}
1	-25	25	-275	275	-275	275
2	25	575	-275	275	-275	-225
3	575	625	-275	275	-275	275
4	625	775	-25	25	225	275
5	625	775	-25	25	-25	25
6	775	825	-275	275	-275	275
7	825	1375	-275	275	-275	-225
8	1375	1425	-275	275	-275	275

Table 7 – Traversable cells of the routing space for the MILP approach.

Instance	Origin						Destination					
	X	Y	Z	Roll	Pitch	Yaw	X	Y	Z	Roll	Pitch	Yaw
1	0	0	275	$-\pi$	0	$-\pi$	300	0	-250	0	$\frac{\pi}{2}$	0
2							600	0	275	0	0	0
3							800	0	275	0	0	0
4							1400	0	275	0	0	0

Table 8 – Origin and destination configurations for the MILP approach.

Instances of the CWRP for the ISA approach

The CWRP instances of the experiments on the ISA approach are illustrated on Figure 10.13. They use the routing space described on Table 7 but also take into account the obstacles described in Table 10. All instances use the same origin and destination configurations presented on Table 9. As previously, the rotation angles that define the orientations are expressed in the roll-pitch-yaw convention.

Instance	Origin						Destination					
	X	Y	Z	Roll	Pitch	Yaw	X	Y	Z	Roll	Pitch	Yaw
1-10	0	0	275	$-\pi$	0	$-\pi$	1400	0	275	0	0	0

Table 9 – Origin and destination configurations for the ISA approach.

Obstacle	X_{min}	X_{max}	Y_{min}	Y_{max}	Z_{min}	Z_{max}	Instance									
							1	2	3	4	5	6	7	8	9	10
1	-25	25	-275	-190,38	112,5	162,5		•	•	•	•	•	•	•	•	•
2	-25	25	-148,08	-21,15	112,5	162,5		•	•	•	•	•	•	•	•	•
3	-25	25	21,15	148,08	112,5	162,5		•	•	•	•	•	•	•	•	•
4	-25	25	190,38	275	112,5	162,5		•	•	•	•	•	•	•	•	•
5	-25	25	-211,54	-126,92	-25	25		•	•	•	•	•	•	•	•	•
6	-25	25	-84,62	84,62	-25	25		•	•	•	•	•	•	•	•	•
7	-25	25	126,92	211,54	-25	25		•	•	•	•	•	•	•	•	•
8	-25	25	-275	-190,38	-162,5	-112,5			•	•	•	•	•	•	•	•
9	-25	25	-148,08	-21,15	-162,5	-112,5			•	•	•	•	•	•	•	•
10	-25	25	21,15	148,08	-162,5	-112,5			•	•	•	•	•	•	•	•
11	-25	25	190,38	275	-162,5	-112,5			•	•	•	•	•	•	•	•
12	137,5	187,5	-275	-190,38	-275	-225			•	•	•	•	•	•	•	•
13	137,5	187,5	-148,08	-21,15	-275	-225			•	•	•	•	•	•	•	•
14	137,5	187,5	21,15	148,08	-275	-225			•	•	•	•	•	•	•	•
15	137,5	187,5	190,38	275	-275	-225			•	•	•	•	•	•	•	•
16	275	325	-211,54	-126,92	-275	-225				•	•	•	•	•	•	•
17	275	325	-84,62	84,62	-275	-225				•	•	•	•	•	•	•
18	275	325	126,92	211,54	-275	-225				•	•	•	•	•	•	•
19	412,5	462,5	-275	-190,38	-275	-225					•	•	•	•	•	•
20	412,5	462,5	-148,08	-21,15	-275	-225					•	•	•	•	•	•
21	412,5	462,5	21,15	148,08	-275	-225					•	•	•	•	•	•
22	412,5	462,5	190,38	275	-275	-225					•	•	•	•	•	•
23	575	625	-275	-190,38	-150	-100					•	•	•	•	•	•
24	575	625	-148,08	-21,15	-150	-100					•	•	•	•	•	•
25	575	625	21,15	148,08	-150	-100					•	•	•	•	•	•
26	575	625	190,38	275	-150	-100					•	•	•	•	•	•
26	575	625	-275	-190,38	100	150					•	•	•	•	•	•
28	575	625	-137,5	137,5	100	150					•	•	•	•	•	•
29	575	625	190,38	275	100	150					•	•	•	•	•	•
30	775	825	-275	-190,38	100	150						•	•	•	•	•
31	775	825	-148,08	-21,15	100	150						•	•	•	•	•
32	775	825	21,15	148,08	100	150						•	•	•	•	•
33	775	825	190,38	275	100	150						•	•	•	•	•
33	775	825	-275	-190,38	-150	-100						•	•	•	•	•
35	775	825	-137,5	137,5	-150	-100						•	•	•	•	•
36	775	825	190,38	275	-150	-100						•	•	•	•	•
36	937,5	987,5	-275	-190,38	-275	-225						•	•	•	•	•
38	937,5	987,5	-137,5	137,5	-275	-225						•	•	•	•	•
39	937,5	987,5	190,38	275	-275	-225						•	•	•	•	•
40	1075	1125	-211,54	-126,92	-275	-225							•	•	•	•
41	1075	1125	-84,62	84,62	-275	-225							•	•	•	•
42	1075	1125	126,92	211,54	-275	-225							•	•	•	•
43	1212,5	1262,5	-275	-190,38	-275	-225								•	•	•
44	1212,5	1262,5	-148,08	-21,15	-275	-225								•	•	•
45	1212,5	1262,5	21,15	148,08	-275	-225								•	•	•
46	1212,5	1262,5	190,38	275	-275	-225								•	•	•
47	1375	1425	-275	-190,38	-162,5	-112,5								•	•	•
48	1375	1425	-148,08	-21,15	-162,5	-112,5								•	•	•
49	1375	1425	21,15	148,08	-162,5	-112,5								•	•	•
50	1375	1425	190,38	275	-162,5	-112,5								•	•	•
51	1375	1425	-211,54	-126,92	-25	25									•	•
52	1375	1425	-84,62	84,62	-25	25									•	•
53	1375	1425	126,92	211,54	-25	25									•	•
54	1375	1425	-275	-190,38	112,5	162,5										•
55	1375	1425	-148,08	-21,15	112,5	162,5										•
56	1375	1425	21,15	148,08	112,5	162,5										•
57	1375	1425	190,38	275	112,5	162,5										•

Table 10 – Obstacles taken into account for the ISA approach.

Bend catalogues

The bend catalogues used in the experiments are presented in Table 11.

Axis	α_b *(in °)*	ρ_b *(in mm)*	L_b *(in mm)*	$B_{cat.}^{90°}$	$B_{cat.}^{45°}$	$B_{cat.}^{30°}$
X (H-bend)	-90	15.0	15.0	•	•	•
	-60		8.660			•
	-45		6.213		•	•
	-30		4.019			•
	30		4.019			•
	45		6.213		•	•
	60		8.660			•
	90		15.0	•	•	•
Y (E-bend)	-90	10.3	10.3	•	•	•
	-60		5.947			•
	-45		4.266		•	•
	-30		2.760			•
	30		2.760			•
	45		4.266		•	•
	60		5.947			•
	90		10.3	•	•	•
Z (Twist)	90	NA	50.0	•	•	•

Table 11 – Bend catalogues for gauge WR75.

Cost coefficients

The waveguide lineic cost and the bend cost used in the experiments are the following:

$$\begin{aligned} \mu &= 1 \\ \gamma_b &= 100 \quad \forall b \in B_{cat.} \mid b \neq \text{Twist} \\ \gamma_b &= 1000 \quad \forall b \in B_{cat.} \mid b = \text{Twist} \end{aligned}$$

Additional proofs

This chapter provides a mathematical demonstration that Constraints 9.13 and 9.14, introduced in Section 9.2.2, force all intermediate bends to be neutral when there is a straight section between the k^{th} and k'^{th} points of neutral fibre $\mathcal{F}_\pi$ (that is to say when $z_{k,k'} = 1$).

Lemma 1:

Let k be an index in $[\![1, K+1]\!]$ and k' be another index in $[\![k+1, K+2]\!]$. If $z_{k,k'} = 1$ and if $z_{n,n'} = 0$ for all $n \in [\![1, k-1]\!]$ and $n' \in [\![k+1, k'-1]\!]$, then:

$$\sum_{\substack{r \in \mathcal{R}_{k''-1} \\ b_r = b_{neut.}}} x_{k''-1,r} = 1 \quad \forall k'' \in [\![k+1, k'-1]\!]$$

Proof: *Let n' be an index in $[\![2, K+1]\!]$. According to Constraints 9.14, a neutral bend must be applied at the n'^{th} point of neutral fibre $\mathcal{F}_\pi$ if $z_{n,n'} = 0$ for all $n \in [\![1, n'-1]\!]$.*
Lemma 1 is obvious when $k' = k+1$ since there are no intermediate bends. The case $k' > k+1$ can be proved by induction. Let k be an index in $[\![1, K]\!]$ and k' be another index in $[\![k+2, K+2]\!]$. Assume that $z_{k,k'} = 1$ and $z_{n,n'} = 0$ for all $n \in [\![1, k-1]\!]$ and $n' \in [\![k+1, k'-1]\!]$. According to Constraint 9.13 for index k, one can verify $z_{k,n'} = 0$ for all $n' \in [\![k+1, K+2]\!]$, therefore it results that $z_{n,n'} = 0$ for all $n \in [\![1, k]\!]$ and $n' \in [\![k+1, k'-1]\!]$. As a consequence, a neutral bend must be applied at the $k+1^{th}$ point of neutral fibre $\mathcal{F}_\pi$.
Now, assume that neutral bends are applied up to the k''^{th} point, with $k+1 \leq k'' < k'-1$. According to Constraints 9.13, one can verify $z_{n,k''+1} = 0$ for all $n \in [\![k+1, k'']\!]$. Using the assumptions of Lemma 1, it can be shown that $z_{n,k''+1} = 0$ for all $n \in [\![1, k'']\!]$. Thus, a neutral bend must also be applied at the $k''+1^{th}$ point of neutral fibre $\mathcal{F}_\pi$. By induction, all intermediate bends between the k^{th} and k'^{th} points of neutral fibre $\mathcal{F}_\pi$ are neutral. In the borderline case where $k'' = k'-1$, the condition $z_{n,k''+1} = 0$ for all $n \in [\![1, k'']\!]$ does not apply since $z_{k,k'} = 1$, which stops the induction.

Property 12: Intermediate neutral bends

Let k be an index in $[\![1, K]\!]$ and k' be another index in $[\![k+1, K+2]\!]$. If $z_{k,k'} = 1$ then:

$$\sum_{\substack{r \in \mathcal{R}_{k''-1} \\ b_r = b_{neut.}}} x_{k''-1,r} = 1 \quad \forall k'' \in [\![k+1, k'-1]\!]$$

Proof: *Property 12 is obvious when* $k' = k + 1$ *since there are no intermediate bends. The case* $k' > k + 1$ *can be proved by induction. Let* k' *be an index in* $[\![3, K + 2]\!]$*. Suppose that* $z_{1,k'} = 1$*. The conditions of Lemma 1 are satisfied since there is no index* n *such that* $n \leq 0$*. Therefore, according to Lemma 1, all intermediate bends between the* 1^{rst} *and* k'^{th} *points of neutral fibre* $\mathcal{F}_\pi$ *are neutral. Now, assume that Property 12 holds up to an index* $k \geq 1$*. Let* k' *be an index in* $[\![k + 2, K + 2]\!]$*. Suppose that* $z_{k+1,k'} = 1$*. Let* n *be an index in* $[\![1, k]\!]$ *and* n' *be another index in* $[\![k + 2, k' - 1]\!]$*. Then* $z_{n,n'} = 0$ *as otherwise the initial assumption, where Property 12 is true for indexes in* $[\![1, k]\!]$*, implies that a neutral bend is applied at the* $k + 1^{th}$ *point of neutral fibre* $\mathcal{F}_\pi$*. However, this is not compatible with* $z_{k+1,k'} = 1$ *and Constraint 9.13 for index* $k + 1$*. Finally, using Lemma 1, all intermediate bends between the* $k + 1^{th}$ *and* k'^{th} *points of neutral fibre* $\mathcal{F}_\pi$ *are neutral. By induction, Property 12 holds for any* $k \in [\![1, K]\!]$ *and* $k' \in [\![k + 2, K + 2]\!]$*.*

Bibliography

[1] Sandip Aine, PP Chakrabarti, and Rajeev Kumar. AWA*-A Window Constrained Anytime Heuristic Search Algorithm. In *IJCAI*, pages 2250–2255, 2007.

[2] Yuto Ando and Hajime Kimura. An automatic piping algorithm including elbows and bends. *Journal of the Japan Society of Naval Architects and Ocean Engineers*, 15:219–226, 2012.

[3] Apache Software Foundation. Apache Commons Math.

[4] Andi Asmara. Automatic piping system implementation: A real case. In *6th international conference on computer and IT applications in the maritime industries, COMPIT'07, Cortona, 23-25 april 2007, U. Nienhuis*, 2007.

[5] Andi Asmara and Ubald Nienhuis. Automatic piping system in ship. In *International Conference on Computer and IT Application (COMPIT)*. Citeseer, 2006.

[6] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

[7] Gleb Belov, Tobias Czauderna, Amel Dzaferovic, Maria Garcia de la Banda, Michael Wybrow, and Mark Wallace. An optimization model for 3D pipe routing with flexibility constraints. In *International Conference on Principles and Practice of Constraint Programming*, pages 321–337. Springer, 2017.

[8] Gleb Belov, Wenbo Du, M Garcia de al Banda, Daniel Harabor, Sven Koenig, and Xinrui Wei. From Multi-Agent Pathfinding to 3D Pipe Routing. In *Symposium on Combinatorial Search (SoCS)*, 2020.

[9] Fawzi Cheik Bessaih. *Optimisation et validation de la charge utile d'un satellite de télécommunication*. PhD thesis, Université d'Avignon, 2013.

[10] Roberto Bisiani. Beam search. *Encyclopedia of Artificial Intelligence*, 2:1467–1468, 1992.

[11] Robert Bridson. Fast Poisson disk sampling in arbitrary dimensions. *SIGGRAPH sketches*, 10(1), 2007.

[12] Ethan Burns and Wheeler Ruml. Iterative-deepening search with on-line tree size prediction. *Annals of Mathematics and Artificial Intelligence*, 69(2):183–205, 2013.

[13] Jean-Thomas Camino, Christian Artigues, Laurent Houssin, and Stéphane Mourgues. A Decomposition Method for Frequency Assignment in Multibeam Satellite Systems. In *International Conference on Operations Research and Enterprise Systems (ICORES)*, page 11P, 2015.

[14] John Canny and John Reif. New lower bound techniques for robot motion planning problems. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 49–60. IEEE, 1987.

[15] Partha Pratim Chakrabarti, Sujoy Ghose, Arup Acharya, and SC De Sarkar. Heuristic search in restricted memory. *Artificial intelligence*, 41(2):197–221, 1989.

[16] Timothy M Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16(4):361–368, 1996.

[17] L. Paul Chew. Constrained Delaunay triangulations. *Algorithmica*, 4(1):97–108, 1989.

[18] Joonsoo Choi, Jürgen Sellen, and Chee-Keng Yap. Approximate Euclidean shortest paths in 3-space. *International Journal of Computational Geometry & Applications*, 7(04):271–295, 1997.

[19] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Breadth-First Search. In *Introduction to Algorithms*. MIT press and McGraw-Hill, 3$^{\text{th}}$ edition, 2009.

[20] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Depth-First Search. In *Introduction to Algorithms*. MIT press and McGraw-Hill, 3$^{\text{th}}$ edition, 2009.

[21] Bence Cserna, William Doyle, Jordan Ramsdell, and Wheeler Ruml. Avoiding dead ends in real-time heuristic search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[22] Michael Cui, Daniel Damir Harabor, and Alban Grastien. Compromise-free Pathfinding on a Navigation Mesh. In *IJCAI*, pages 496–502, 2017.

[23] Rina Dechter and Judea Pearl. Generalized Best-First Search strategies and the optimality of a*. *Journal of the ACM (JACM)*, 32(3):505–536, 1985.

[24] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[25] James E Doran and Donald Michie. Experiments with the graph traverser program. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 294(1437):235–259, 1966.

[26] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.

[27] Russell Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 39–43. Ieee, 1995.

[28] Xiaoning Fan, Yan Lin, and Zhuoshang Ji. The ant colony optimization for ship pipe route design in 3D space. In *2006 6th World Congress on Intelligent Control and Automation*, volume 1, pages 3103–3108. IEEE, 2006.

[29] David Furcy and Sven Koenig. Limited discrepancy beam search. In *IJCAI*, 2005.

[30] David Furcy and Sven Koenig. Scaling up WA* with Commitment and Diversity. In *IJCAI*, pages 1521–1522, 2005.

[31] Marcus Furuholmen, Kyrre Glette, Mats Hovin, and Jim Torresen. Evolutionary approaches to the three-dimensional multi-pipe routing problem: a comparative study using direct encodings. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 71–82. Springer, 2010.

[32] Gerald Gamrath, Daniel Anderson, Ksenia Bestuzheva, Wei-Kun Chen, Leon Eifler, Maxime Gasse, Patrick Gemander, Ambros Gleixner, Leona Gottwald, Katrin Halbig, Gregor Hendel, Christopher Hojny, Thorsten Koch, Pierre Le Bodic, Stephen J. Maher, Frederic Matter, Matthias Miltenberger, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Christine Tawfik, Stefan Vigerske, Fabian Wegscheider, Dieter Weninger, and Jakob Witzig. The SCIP Optimization Suite 7.0. Technical report, Optimization Online, March 2020.

[33] Gerald Gamrath, Daniel Anderson, Ksenia Bestuzheva, Wei-Kun Chen, Leon Eifler, Maxime Gasse, Patrick Gemander, Ambros Gleixner, Leona Gottwald, Katrin Halbig, Gregor Hendel, Christopher Hojny, Thorsten Koch, Pierre Le Bodic, Stephen J. Maher, Frederic Matter, Matthias Miltenberger, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Christine Tawfik, Stefan Vigerske, Fabian Wegscheider, Dieter Weninger, and Jakob Witzig. The SCIP Optimization Suite 7.0. ZIB-Report 20-10, Zuse Institute Berlin, March 2020.

[34] Joseph L Ganley and James P Cohoon. Routing a multi-terminal critical net: Steiner tree construction in the presence of obstacles. In *Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS'94*, volume 1, pages 113–116. IEEE, 1994.

[35] Reginaldo Guirardello and Ross E Swaney. Optimization of process plant layout with pipe routing. *Computers & chemical engineering*, 30(1):99–114, 2005.

[36] Daniel Damir Harabor, Alban Grastien, Dindar Öz, and Vural Aksakalli. Optimal any-angle pathfinding in practice. *Journal of Artificial Intelligence Research*, 56:89–118, 2016.

[37] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[38] William D Harvey and Matthew L Ginsberg. Limited discrepancy search. In *IJCAI (1)*, pages 607–615, 1995.

[39] John Hershberger and Subhash Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM Journal on Computing*, 28(6):2215–2256, 1999.

[40] David W Hightower. A solution to line-routing problems on the continuous plane. In *Proceedings of the 6th annual Design Automation Conference*, pages 1–24, 1969.

[41] Jörg Hoffmann. Extending ff to numerical state variables. In *ECAI*, pages 571–575. Citeseer, 2002.

[42] Yong K Hwang and Narendra Ahuja. Gross motion planning—a survey. *ACM Computing Surveys (CSUR)*, 24(3):219–291, 1992.

[43] Satoshi Ikehira, Hajime Kimura, Eisuke Ikezaki, and Hiroyuki Kajiwara. Automatic design for pipe arrangement using multi-objective genetic algorithms. *Journal of the Japan Society of Naval Architects and Ocean Engineers*, 2:155–160, 2005.

[44] Teruaki Ito. Piping layout wizard: basic concepts and its potential for pipe route planning. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 438–447. Springer, 1998.

[45] Teruaki Ito. A genetic algorithm approach to piping route path planning. *Journal of Intelligent Manufacturing*, 10(1):103–114, 1999.

[46] Teruaki Ito. Route planning wizard: basic concept and its implementation. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 547–556. Springer, 2002.

[47] Wen-Ying Jiang, Yan Lin, Ming Chen, and Yan-Yun Yu. A co-evolutionary improved multi-ant colony optimization for ship multiple and branch pipe route design. *Ocean Engineering*, 102:63–70, 2015.

[48] Shi Kai-jian and Zhu Hong-e. Efficient routing algorithm. *Computer-aided design*, 19(7):375–379, 1987.

[49] Hermann Kaindl and Aliasghar Khorsand. Memory-bounded bidirectional search. In *AAAI*, pages 1359–1364, 1994.

[50] Sang-Seob Kang, Sehyun Myung, and Soon-Hung Han. A design expert system for auto-routing of ship pipes. *Journal of Ship Production*, 15(01):1–9, 1999.

[51] Sanjiv Kapoor and SN Maheshwari. Efficient algorithms for Euclidean shortest path and visibility problems with polygonal obstacles. In *Proceedings of the fourth annual symposium on computational geometry*, pages 172–182, 1988.

[52] Sanjiv Kapoor, SN Maheshwari, and Joseph SB Mitchell. An efficient algorithm for Euclidean shortest paths among polygonal obstacles in the plane. *Discrete & Computational Geometry*, 18(4):377–383, 1997.

[53] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.

[54] Shin-Hyung Kim, Won-Sun Ruy, and Beom Seon Jang. The development of a practical pipe auto-routing system in a shipbuilding CAD environment using network optimization. *International journal of naval architecture and ocean engineering*, 5(3):468–477, 2013.

[55] Hajime Kimura. Automatic designing system for piping and instruments arrangement including branches of pipes. In *International Conference on Computer Applications in Shipbuilding (ICCAS)*, pages 93–99, 2011.

[56] David G Kirkpatrick and Raimund Seidel. The ultimate planar convex hull algorithm? *SIAM journal on computing*, 15(1):287–299, 1986.

[57] Sven Koenig and Xiaoxun Sun. Comparing real-time and incremental heuristic search for real-time situated agents. *Autonomous Agents and Multi-Agent Systems*, 18(3):313–341, 2009.

[58] Richard E Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial intelligence*, 27(1):97–109, 1985.

[59] C. Y. Lee. An algorithm for path connections and its applications. *IRE Transactions on Electronic Computers*, EC-10(3):346–365, 1961.

[60] Der-Tsai Lee and Franco P Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14(3):393–410, 1984.

[61] Thomas Lengauer. *Combinatorial algorithms for integrated circuit layout*. Springer Science & Business Media, 2012.

[62] Maxim Likhachev, Geoffrey J Gordon, and Sebastian Thrun. Ara*: Anytime a* with provable bounds on sub-optimality. *Advances in neural information processing systems*, 16:767–774, 2003.

[63] Lijia Liu and Qiang Liu. Multi-objective routing of multi-terminal rectilinear pipe in 3D space by MOEA/D and RSMT. In *2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 462–467. IEEE, 2018.

[64] Benachir Medjdoub and Gang Bi. Parametric-based distribution duct routing generation using constraint-based design approach. *Automation in Construction*, 90:104–116, 2018.

[65] Koichi Mikami. A computer program for optimal routing of printed circuit connectors. *IFIPS Proc., 1968*, 1968.

[66] Edward F Moore. The shortest path through a maze. In *Proceedings of the International Symposium on the Theory of Switching*, pages 285–292, 1959.

[67] Alex Nash, Kenny Daniel, Sven Koenig, and Ariel Felner. Theta*: Any-angle path planning on grids. In *AAAI*, volume 7, pages 1177–1183, 2007.

[68] Alex Nash, Sven Koenig, and Craig Tovey. Lazy theta*: Any-angle path planning and path length analysis in 3d. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, 2010.

[69] Wentie Niu, Haiteng Sui, Yaxiao Niu, Kunhai Cai, and Weiguo Gao. Ship pipe routing design using NSGA-II and coevolutionary algorithm. *Mathematical Problems in Engineering*, 2016, 2016.

[70] Christos H Papadimitriou. An algorithm for shortest-path motion in three dimensions. *Information processing letters*, 20(5):259–263, 1985.

[71] Jin-Hyung Park and Richard L Storch. Pipe-routing algorithm development: case study of a ship engine room design. *Expert Systems with Applications*, 23(3):299–309, 2002.

[72] Judea Pearl and Jin H. Kim. Studies in Semi-Admissible Heuristics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(4):392–399, 1982.

[73] Laurent Perron and Vincent Furnon. OR-Tools.

[74] Ira Pohl. First results on the effect of error in heuristic search. *Machine Intelligence*, 5:219–236, 1970.

[75] Ira Pohl. Bi-Directional Search. In *Machine Intelligence*, volume 6, pages 127–140. Edinburgh University Press, bernard meltzer and donald michie edition, 1971.

[76] Yanfeng Qu, Dan Jiang, and Qingyan Yang. Branch pipe routing based on 3d connection graph and concurrent ant colony optimization algorithm. *Journal of Intelligent Manufacturing*, 29(7):1647–1657, 2018.

[77] Alexander Reinefeld and T. Anthony Marsland. Enhanced iterative-deepening search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7):701–710, 1994.

[78] Elaine Rich and Kevin Knight. *Artificial Intelligence: Instructor's Manual*. McGraw-Hill, 1992.

[79] H Richert and G Gruhn. A numeric-heuristic system for plant wide pipe routing. *Computers & Chemical Engineering*, 23:S735–S738, 1999.

[80] Eliott Roynette. *Optimisation de la conception du design du harnais de commande des véhicules spatiaux*. PhD thesis, Toulouse, ISAE, 2018.

[81] Stuart Russell and Peter Norvig. Artificial Intelligence: a modern approach. pages 75–81. Prentice Hall, 3$^{\text{th}}$ edition, 2002.

[82] Stuart J.d Russell. Efficient Memory-Bounded Search Methods. In *ECAI*, volume 92, pages 1–5, 1992.

[83] Abdelilah Sakti, Lawrence Zeidner, Tarik Hadzic, Brian St Rock, and Giusi Quartarone. Constraint programming approach for spatial packaging problem. In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 319–328. Springer, 2016.

[84] H Schmidt-Traub, M Köster, T Holtkötter, and N Nipper. Conceptual plant layout. *Computers & chemical engineering*, 22:S499–S504, 1998.

[85] Henner Schmidt-Traub, Torsten Holtkötter, Michael Lederhose, and Peter Leuders. An approach to plant layout optimization. *Chemical Engineering & Technology: Industrial Chemistry-Plant Equipment-Process Engineering-Biotechnology*, 22(2):105–109, 1999.

[86] Micha Sharir and Amir Schorr. On shortest paths in polyhedral spaces. *SIAM Journal on Computing*, 15(1):193–215, 1986.

[87] JIRI Soukup. Fast maze router. In *Design Automation Conference*, pages 100–101. IEEE Computer Society, 1978.

[88] Marvin Stanczak, Cédric Pralet, Vincent Vidal, and Vincent Baudoui. Optimal Pipe Routing Techniques in an Obstacle-Free 3D Space. In *Proceedings of the 9th International Conference on Operations Research and Enterprise Systems (ICORES)*, volume 1, pages 69–79. INSTICC, SciTePress, 2020.

[89] Marvin Stanczak, Cédric Pralet, Vincent Vidal, and Vincent Baudoui. Techniques optimales pour le routage de canalisations dans un espace 3D sans obstacle. In *ROADEF 2020 (Congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision)*, 2020.

[90] Marvin Stanczak, Cédric Pralet, Vincent Vidal, and Vincent Baudoui. A Pipe Routing Hybrid Approach based on A* Search and Linear Programming. In *Proceedings of the 18th International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR)*, pages 179–195. Springer, 2021.

[91] Marvin Stanczak, Cédric Pralet, Vincent Vidal, and Vincent Baudoui. Une approche hybride de routage de canalisation basée sur la recherche A* et la programmation linéaire. In *ROADEF 2021 (Congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision)*, 2021.

[92] Haiteng Sui and Wentie Niu. Branch-pipe-routing approach for ships using improved genetic algorithm. *Frontiers of Mechanical Engineering*, 11(3):316–323, 2016.

[93] Jordan T Thayer, Wheeler Ruml, and Ephrat Bitton. Fast and loose in bounded suboptimal heuristic search. In *Proceedings of the First International Symposium on Search Techniques in Artificial Intelligence and Robotics (STAIR-08)*, 2008.

[94] Jordan Tyler Thayer and Wheeler Ruml. Faster than Weighted A*: An Optimistic Approach to Bounded Suboptimal Search. In *ICAPS*, pages 355–362, 2008.

[95] Christian Van der Velden, Cees Bil, Xinghuo Yu, and Adrian Smith. Towards A Knowledge Based Cable Router for Aerospace Vehicles. In *IKE*, pages 95–101, 2006.

[96] Christian Van der Velden, Cees Bil, Xinghuo Yu, and Adrian Smith. An intelligent system for automatic layout routing in aerospace design. *Innovations in Systems and Software Engineering*, 3(2):117–128, 2007.

[97] Christian Van der Velden, Cees Bil, Xinghuo Yu, and Adrian Smith. An intelligent system for routing automation. In *Innovative Production Machines and Systems, Virtual Conference*, pages 2–13, 2008.

[98] Samuel Vogel and Stephan Rudolph. Automated piping with standardized bends in complex systems design. In *International Conference on Complex Systems Design & Management*, pages 113–124. Springer, 2016.

[99] Christopher Makoto Wilt, Jordan Tyler Thayer, and Wheeler Ruml. A Comparison of Greedy Search Algorithms. In *3rd Annual Symposium on Combinatorial Search*, 2010.

[100] Weixiong Zhang. Complete anytime beam search. In *AAAI/IAAI*, pages 425–430, 1998.

[101] Rong Zhou and Eric A Hansen. Memory-Bounded A* Graph Search. In *FLAIRS conference*, pages 203–209, 2002.

[102] Rong Zhou and Eric A Hansen. Multiple Sequence Alignment Using Anytime A*. In *AAAI/IAAI*, pages 975–977, 2002.

[103] Rong Zhou and Eric A Hansen. Beam-Stack Search: Integrating Backtracking with Beam Search. In *ICAPS*, pages 90–98, 2005.

[104] David Zhu and J-C Latombe. Pipe routing-path planning (with many constraints). In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 1940–1941. IEEE Computer Society, 1991.