



HAL
open science

A Robust and Efficient Method for Solving Geometrical Constraint Problems by Homotopy

Rémi Imbach, Pascal Mathis, Pascal Schreck

► **To cite this version:**

Rémi Imbach, Pascal Mathis, Pascal Schreck. A Robust and Efficient Method for Solving Geometrical Constraint Problems by Homotopy. [Research Report] RR-8705, INRIA. 2015. hal-01135230v1

HAL Id: hal-01135230

<https://inria.hal.science/hal-01135230v1>

Submitted on 26 Mar 2015 (v1), last revised 25 Jul 2016 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A Robust and Efficient Method for Solving Geometrical Constraint Problems by Homotopy

Rémi Imbach, Pascal Mathis, Pascal Schreck

**RESEARCH
REPORT**

N° 8705

Mars 2015

Project-Team VEGAS



A Robust and Efficient Method for Solving Geometrical Constraint Problems by Homotopy

Rémi Imbach*, Pascal Mathis†, Pascal Schreck†

Project-Team VEGAS

Research Report n° 8705 — Mars 2015 — 35 pages

Abstract: The goal of Geometric Constraint Solving is to find 2D or 3D placements of some geometric primitives fulfilling some geometric constraints. The common guideline is to solve them by a numerical iterative method (e.g. Newton-Raphson method). A sole solution is obtained whereas many exist. But the number of solutions can be exponential and methods should provide solutions close to a sketch drawn by the user. Assuming that a decomposition-recombination planner is used, we consider irreducible problems. Geometric reasoning can help to simplify the underlying system of equations by changing a few equations and triangularizing it. This triangularization is a geometric construction of solutions, called construction plan. We aim at finding several solutions close to the sketch on a one-dimensional path defined by a global parameter-homotopy using the construction plan. Some numerical instabilities may be encountered due to specific geometric configurations. We address this problem by changing on-the-fly the construction plan. Numerical results show that this hybrid method is efficient and robust.

Key-words: Geometric Constraint Solving Problems, Reparameterization, Curve Tracking, Symbolic-Numeric Algorithm

* Travail en partie effectué au laboratoire ICube.

† Laboratoire ICube - UMR 7357 CNRS Université de Strasbourg, bd Sébastien Brant, F-67412 Illkirch Cedex, France

**RESEARCH CENTRE
NANCY – GRAND EST**

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Une méthode robuste et efficace pour résoudre des systèmes de contraintes géométriques par homotopie

Résumé : Le but de la résolution de problèmes de contraintes géométriques est de produire des figures 2D ou 3D satisfaisant un ensemble de contraintes portant sur leur géométrie. De tels problèmes sont en général résolus grâce à une méthode numérique, souvent Newton-Raphson, qui ne produit qu'une solution alors qu'il en existe un nombre exponentiel. Celles ressemblant à l'esquisse sont d'un intérêt particulier. Les approches par décomposition-assemblage produisent des systèmes irréductibles qui peuvent comporter beaucoup de contraintes. Le raisonnement géométrique peut cependant aider à simplifier les systèmes d'équations correspondant en remplaçant quelques équations, ce qui permet de les triangulariser. Une telle triangularisation est une construction géométrique des solutions et est appelée plan de construction. On se propose dans ce rapport de trouver plusieurs solutions, proches de l'esquisse sur une courbe définie par une homotopie utilisant le plan de construction pour réduire son coût. L'utilisation d'un plan de construction induit des instabilités numériques à proximité de certains points; ces instabilités sont évitées en changeant le plan de construction pendant le suivi de la courbe. La méthode décrite ici a été implémentée, et les résultats obtenus montrent son efficacité et sa robustesse.

Mots-clés : Problèmes de résolution de contraintes géométriques, Re-paramétrisation, Suivi de courbes, Algorithme symbolique-numérique

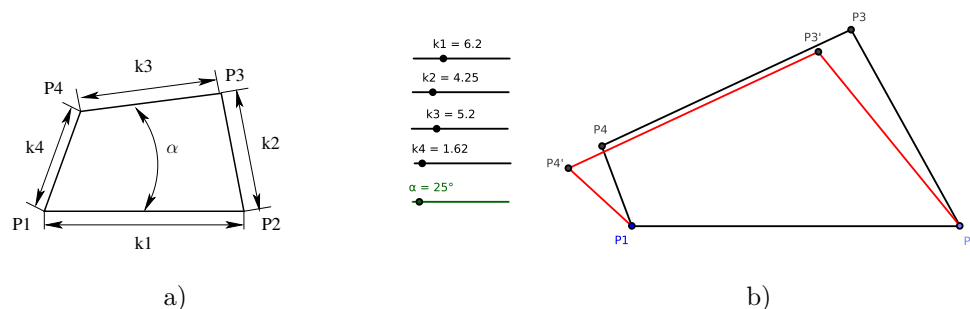


Figure 1: Dimensioned sketch and two close solutions

1 Introduction

Geometric Constraints Solving Problems (GCSP) arise in many fields such as CAD, robotics or molecular modeling. This is to determine the positions of geometric elements (points, lines, planes, circles, etc.) that must satisfy a set of constraints such as distances, angles, tangencies and so on. Commercial solvers generally rely on numerical methods (Newton or quasi-Newton) that provide a single solution. But even when problems are well-constrained, there may be several interesting solutions. Example of Fig. 1 shows a 2D sketch drawn by a user who imposed five constraints on it: 4 distances and 1 angle. For some assignments, there could be two solutions very close to each other (Fig. 1b) and there is no reason for excluding one from another. However, providing all the solutions is not always relevant since the number of solutions can be significant. In a 2D triangle “mesh” with n well-constrained triangles, a complete solving leads to 2^n solutions, some of which are very far from the shape of the sketch. Our aim is to design a method that uses the sketch to guide the research of several solutions. We assume that the considered GCSP are structurally well-constrained. Roughly speaking, this means that there exists some assignments for dimensions leading to finitely many solutions.

Several methods can yield more than one and even all the solutions. Subdivision methods [15, 5] give good results but the number of boxes to be explored can be huge. In algebraic approaches, homotopy methods have been successfully studied in this area [3] but only for small size problems. Indeed, the number of homotopy paths to follow grows exponentially with the number of constraints. [11] proposes to use the sketch to define a parameter-homotopy. Thus a sole path is followed but a sole solution is obtained.

Another way to get several solutions comes from geometric methods. Using geometric construction rules a construction plan is produced by the solver. The construction plan is evaluated numerically to yield different solutions. However, some problems in 2D and most 3D problems can not be solved by a sequence of elementary geometric constructions. To circumvent this, [6] proposes a reparameterization: constraints are replaced by some constraints of distances or angles in order to make possible the production of a symbolic construction plan. Next, the numerical resolution must find the values of added constraints to satisfy the deleted constraints. Again, the choice of the numerical resolution method is crucial to provide several solutions. [8] presents a first attempt for using a homotopy method along with a parameterized construction plan. The idea is to follow a homotopy path to which belongs the sketch.

In [9] it is shown that this path may contain several solutions and that parameterized construction plans can be used to detect solutions at infinity. In addition, the construction plan can be used to generate new sketches and thus new paths.

By combining homotopy and reparameterization, the number of unknowns of the problem is

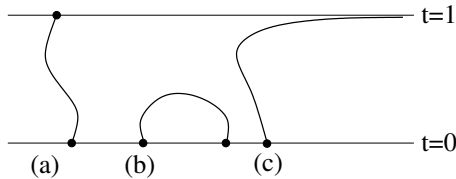


Figure 2: Different topologies of homotopy paths crossing \mathcal{P}_0 .

greatly reduced and these unknowns apply only to added constraints. The solution is structured in a construction plan, and the price to pay is that this plan is much longer to evaluate than polynomials of constraints. This method provides very good results, both in execution time and number of solutions found, but is unstable at the boundary of the domain. The boundary corresponds to geometric situations which are clearly identified such as a circle-circle intersection where the circles are almost tangent. In these situations, numerical path tracking is difficult.

This paper justifies the resolution by homotopy of a reparameterized construction plan. It defines the boundary of the domain of definition and characterizes it at specific geometric configurations. The originality of this work is a change of the construction plan on-the-fly to avoid getting too close to the boundary. In this way we solve the stability problems of this method.

This paper is structured as follows. Section 2 reviews homotopy continuation methods and outlines our method on an example. Section 3 introduces definitions and basic concepts on construction plans. Section 4 presents homotopy paths tracking on construction plan. Section 5 explains how to change a construction plan on-the-fly to avoid critical situations. Section 6 presents some results and section 7 concludes this work.

2 Homotopy solving

2.1 A brief on classical homotopy methods

Homotopy methods use continuation approach to find zero values of $F : \mathbb{K}^m \rightarrow \mathbb{K}^m$ by embedding it in an homotopy function H that interpolates an initial function $F_0 : \mathbb{K}^m \rightarrow \mathbb{K}^m$ into F . H can for instance be defined as $H(x, t) = (1 - t)F(x) + tF_0(x)$. When H is assumed to be regular and defined on $\mathbb{K}^m \times [0, 1]$, connected components of the set $H^{-1}(0) = \{(x, t) \in \mathbb{K}^m \times [0, 1] \mid H(x, t) = 0\}$ are one dimensional manifolds which borders are elements of hyperplanes $\mathcal{P}_i = \{(x, t) \in \mathbb{K}^m \times \mathbb{R} \mid t = i\}$ for $i \in \{0, 1\}$. These connected components are called *homotopy paths* and are diffeomorphic to a close, a semi-open or an open interval, or a circle.

If a zero value x_0 of F_0 is known, a homotopy paths \mathcal{S} is followed from $(x_0, 0)$ by a path tracking method, until a zero-value of F is found provided that (a) \mathcal{S} is diffeomorphic to $[0, 1]$ and has its other extremity in \mathcal{P}_1 . Bad cases that can occur in a general continuation framework are the following: (b) \mathcal{S} is diffeomorphic to $[0, 1]$ but has its other extremity in \mathcal{P}_0 and no zero values of F is found, or (c) \mathcal{S} is diffeomorphic to $[0, 1[$, has infinite length and the tracking process does not terminate (Fig. 2 shows these different cases).

Most of literature concerning homotopy methods addresses situations where $\mathbb{K} = \mathbb{C}$, and components of F are polynomials; see [17] for a survey. We introduce here some selected points.

When $\mathbb{K} = \mathbb{C}$ is assumed, it is possible to ensure with probability one that H is regular by using the so-called *gamma-trick* and that homotopy paths of H are strictly increasing with respect to the interpolation parameter t . Hence case (b) can not occur.

When components of F are polynomials one can construct F_0 having at least as many roots as

Unknowns:
 point P_0, \dots, P_5
Parameters:
 length h_0, \dots, h_8
Constraints:
 $distance(P_0, P_1) = h_0$
 $distance(P_1, P_2) = h_1$
 ...
 $distance(P_1, P_4) = h_7$
 $distance(P_2, P_5) = h_8$

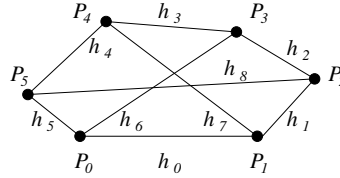


Figure 3: A symbolic statement (left part) and a dimensioned sketch (right part) of the GCSP $K_{3,3}$. Edges are distances constraints of parameters h_i .

F . In this case all roots of F are linked to roots of F_0 , and paths (c) cross \mathcal{P}_1 in a point of infinite norm, called *solution at infinity*. Such paths can be detected before following them in a complex projective space by using an homogenization of H . Finding all complex roots of a polynomial map F by homotopy is then conditioned to the ability of constructing an initial polynomial map F_0 admitting as many known roots as F . The number of roots of F is bounded by the total degree of F , but often over-estimated from far by the latter, especially when polynomials are sparse and structured. Starting an homotopy with this initial system leads to follow too much path. The bound of the number of roots can be sharpened by exploiting an homogeneous structure of F (see [12]) or thanks to the Bernstein’s theorem [7].

A survey on path tracking processes can be found in [1]. The most common one uses a prediction correction approach: knowing an initial (approximated) point (x_0, t_0) on \mathcal{S} , a predictor compute a new approximated point of \mathcal{S} , and a corrector brings it much closer to the path. Prediction is often done along the tangent T on (x_0, t_0) of \mathcal{S} with a *prediction step* δ , and the corrector consists of Newton-Raphson iterations on H that has to be squared thanks to an additional equation. The latter can constraint the correction to lie on the hyperplane orthogonal to T . When F is polynomial and \mathcal{S} is proved to be monotonic with respect to t , corrected point is searched such that $t = t_0 + \delta$. The step δ is classically adapted dynamically trough the tracking process. Initialized by [10], recent researches on path tracking attempt to improve robustness and goes trough certification by using clever arithmetic such multi-precision arithmetic ([2]) or interval arithmetic ([10, 4, 13]).

2.2 Using homotopy to solve a GCSP

2.2.1 Translating a GCSP into a system of equations (\mathcal{F})

A GCSP G is denoted by $G = C[X, A]$ where C is a set of constraints, X a set of unknowns and A a set of parameters. Usually, unknowns are geometric entities such as points and lines and parameters are dimensions such as lengths or angles. Constraints are predicative terms such that $distance(p1, p2) = k$ for a distance constraint between points $p1$ and $p2$. A example is given in Fig. 3 where the problem consists in constructing six points P_0, \dots, P_5 in a plane knowing 9 distances pairwise. We call this GCSP $K_{3,3}$ ¹. This definition of GCSP is general and for some statements it could be useful to consider some geometric entities as parameters and some dimensions as unknowns.

¹when considering right part of Fig. 3 as a non-oriented graph, it is the complete bipartite graph with 3 vertex in each component

Considering that a numerical function is associated to each constraint, a GSCP is easily translated into a numerical function where algebraic unknowns are associated to GCSP unknowns. Since point-point distances constraints are invariant up to rigid motions, solutions are sought in a chosen reference; for instance in $K_{3,3}$ we could search figures with P_0 at the origin and P_1 with null ordinate. Algebraic unknowns $\{x_0, \dots, x_8\}$ are assigned to remaining free coordinates of GCSP unknowns: x_0 to the abscissa of P_1 , (x_1, x_2) to the coordinates of P_2 , and so on.

Then, equation $F_0(x_1) = P_0P_1 - h_0$ where P_0P_1 denotes euclidean distance between P_0 and P_1 is associated to distance constraint $distance(P_0, P_1) = h_0$. We will note $F : \mathbb{R}^9 \rightarrow \mathbb{R}^9$ the function whose components are numerical functions associated to constraints of G , and call F the *numerical function* associated to G . Since G is structurally well constrained, F is well-formed: it has as many variables as equations, and no subset of equations contains fewer unknowns than equations.

In a numerical function F , we also denote by X the set $\{x_0, \dots, x_m\}$ of algebraic unknowns, and zeros of F are called *figures*. Since F also depends on parameters A , and we make it explicitly appear by noting

$$F(X, A) = 0. \quad (1)$$

If A_{so} are the values of parameters given by the user, solving G for parameters A_{so} is equivalent to solve the system of unknowns X :

$$F(X, A_{so}) = 0. \quad (2)$$

In homotopy path tracking, a system whose some solutions are known is required. As previously mentioned we will use the sketch as initial solution. Values for X read on the sketch drawn will be referred to as X_{sk} . We have especially

$$F(X_{sk}, A_{sk}) = 0 \quad (3)$$

where A_{sk} is the values of parameters read on the sketch.

2.2.2 Searching zero-values of F by homotopy

When components of F are polynomials, all solutions in \mathbb{C}^m of equation (2) can be provided by a classical homotopy method. [3] proposes to first use symbolic reduction on F to obtain $F' : \mathbb{C}^{m'} \rightarrow \mathbb{C}^{m'}$ with $m' < m$ in order to decrease the number of path to be followed. But most of found solutions are complex whereas in applications only real figures are relevant.

GCSP generally admit a number of solutions that grows exponentially with the number of constraints; in CAD applications, a user may prefer obtaining one or several solutions close to the sketch he drawn (that is supposed to represent well the desired solution) than choosing between an exponential number of figure. With this motivation, [11] uses the sketch to build the homotopy function H : taking (3) into account, H is defined as $H(X, t) = F(X, tA_{so} + (1-t)A_{sk})$, and $(X_{sk}, 0)$ is the sole known solution to $H(X, t) = 0$. Only one homotopy path is followed, in the real space $\mathbb{R}^m \times [0, 1]$, and only one solution is found, but it is considered to be close to the sketch, as a continuous deformation of the latter.

The main limitations of a real homotopy lies in the existence of homotopy paths (b) and (c): following only one path may lead to obtain a solution of $H(X, 0) = 0$ in a finite time, or no solution in an infinite time. In the previous paper [9], we have proposed to re-use the main idea of [11], that is building the homotopy function from the sketch and tracking paths in a real space in order to obtain only real solutions close to the sketch. We have showed how to define the homotopy function in such a way that paths (c) can be detected during the tracking process because figures lying on it converge to a special geometric configuration. Moreover others paths,

considered in $\mathbb{R}^m \times \mathbb{R}$, are showed to be diffeomorphic to circles, and can be tracked until they loop, to obtain more solutions. Finally it shows how to use constructive geometry to find more solutions.

If this approach does not ensure to obtain neither one nor all the solutions of a GCSP, it is efficient in practice and sometimes allows to get all solutions. It provides several solutions to problem that are too big to be solved with a classical homotopy method. For the sake of self-sufficiency we recall here main contributions of [9] on which relies the method presented here. The idea of tracking a path in $\mathbb{R}^m \times \mathbb{R}$ to obtain several solutions on a sole path is not new, and is called *global homotopy* in page 81 of [1].

2.3 Using a global homotopy to solve GCSP

Key results of our global homotopy approach comes from the way the parameters A_{sk} are interpolated into A_{so} . Let us introduce two notions in order to characterize it.

Definition 1 We call *interpolation function* from $\alpha \in \mathbb{R}$ to $\beta \in \mathbb{R}$ a C^∞ function $a : \mathbb{R} \rightarrow \mathbb{R}$ that satisfies $a(0) = \alpha$ and $a(1) = \beta$. Let $A_{sk} = \{A_{sk}^0, \dots, A_{sk}^{p-1}\}$ and $A_{so} = \{A_{so}^0, \dots, A_{so}^{p-1}\}$ be two vectors of \mathbb{R}^p . We say that $a : \mathbb{R} \rightarrow \mathbb{R}^p$ is an *interpolation function* from A_{sk} to A_{so} if, for $0 \leq j \leq p-1$, its component a_j is an interpolation function from A_{sk}^j to A_{so}^j .

With GCSP $K_{3,3}$ of figure 3, with A_{sk} the sketch values and A_{so} the required values, the function $a : \mathbb{R} \rightarrow \mathbb{R}^9$ is made up of components a_i that could be for instance:

$$a_i(t) = \begin{cases} -ct^2 + (A_{so}^i - A_{sk}^i + c)t + A_{sk}^i & \text{if } i = 0 \\ (1-t)A_{sk}^i + tA_{so}^i & \text{otherwise} \end{cases} \quad (4)$$

with $c > 0$ is an interpolation function from A_{sk} to A_{so} .

Definition 2 If a is an interpolation function between $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$, we call *positive support* of a , and note it $Supp(a)$, the set $\{x \in \mathbb{R} | a(x) \geq 0\}$.

The map a_0 defined in equation (4) has a positive support that is equal to the close real interval between the two real roots of a_0 .

If a is an interpolation function from A_{sk} to A_{so} and F a numerical function associated to a GCSP, we define the homotopy function $H : \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^m$ as:

$$H(X, t) = F(X, a(t)). \quad (5)$$

For instance, the homotopy function associated to the GCSP $K_{3,3}$ is $H : \mathbb{R}^9 \times \mathbb{R} \rightarrow \mathbb{R}^9$

$$H : \quad \mathbb{R}^9 \times \mathbb{R} \quad \rightarrow \quad \mathbb{R}^9 \\ x_0, \dots, x_8, t \quad \mapsto \quad \begin{cases} P_0P_1 - a_0(t) \\ P_1P_2 - a_1(t) \\ P_2P_3 - a_2(t) \\ \dots \\ P_2P_5 - a_8(t) \end{cases} \quad (6)$$

where a is defined as in equation (4).

Main contributions of [9] are resumed by the following affirmation. If the following assumptions are met:

- (i) H is well defined over a $m+1$ dimensional manifold $\mathcal{DH} \subseteq \mathbb{R}^m \times \mathbb{R}$ without any border,

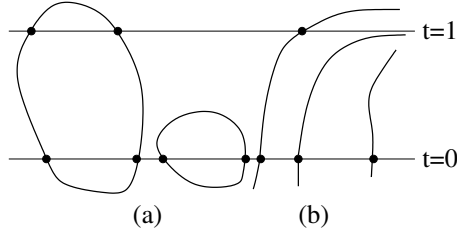


Figure 4: Different topologies of homotopy paths crossing \mathcal{P}_0 possibly encountered by our global homotopy method.

- (ii) H is regular on \mathcal{DH} ,
- (iii) the 0 – th component of H corresponds to a distance constraint and a_0 has a compact positive support,

then connected components of $H^{-1}(0) \in \mathbb{R}^m \times \mathbb{R}$ are either diffeomorphic (a) to a circle, or (b) to an open interval. Figure 4 presents such different paths. If an homotopy path \mathcal{S} is diffeomorphic to an open interval then extremities of \mathcal{S} converge either to a point in $(\mathbb{R}^m \times \mathbb{R}) \setminus \mathcal{DH}$ either to a solution at infinity. These cases correspond to special geometric configuration that can be detected during the tracking process.

Assumption (i) holds when constraints are distance, colinearity, ..., angles between three points. The numerical function associated to the latter is not defined when two of the three points are coincident, and this geometric configuration can characterize a bounded limit of a homotopy path diffeomorphic to an open interval. Assumption (ii) stands for almost all values of A_{sk} and A_{so} , when each constraint depends on a numerical parameter. We also suppose that parameters are independents, that is each parameter is involved in a sole constraint. Assumption (iii) means that G involves at least one distance constraint, this is stronger than necessary but ensures the result to be true.

This gives rise to the simple following procedure, which is parameterized by a path tracking method:

Continuation solving of GCSP

Input: GCSP $C[X, A]$, values A_{so} , sketch X_{sk}

Output: list of solutions \mathcal{L}_{sol}

1. Compute values A_{sk} from X_{sk} , build an interpolation function a from A_{sk} to A_{so} , build H .
2. Follow path of $H^{-1}(0)$ from $(X_{sk}, 0)$ until it loops on $(X_{sk}, 0)$ while checking if :
 - it converges to a solution at infinity, then stop;
 - it converges to a figure outside of \mathcal{DH} , then stop;
 - it crosses hyperplane \mathcal{P}_1 , then append solution to \mathcal{L}_{sol} .

The amount of computation required by the step 2 to follow a path \mathcal{S} depends of course on the chosen prediction step and on the arc-length, but is also linked to the cost of an iteration. If the path tracker that is used is a simple predictor-corrector with a first-order predictor and a correction performed by Newton-Raphson iterations ², it requires at each iteration several

²such a path tracker has been used in our experiments and is briefly described in subsection 6.2.1

inversions of an $m \times m$ matrix. The cost of our global homotopy solving is then directly related to the number of components of H .

The point of the present paper is to use constructive geometry to track paths of H in a space of dimension d , with $d \ll m$.

2.4 Using reparameterization to quasi-triangularize the system of equations

The presentation is made with example $K_{3,3}$, but what is said is directly generalizable to others GCSP.

Consider the function H^R defined as follows:

$$H^R : \mathbb{R}^9 \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^{10} \\ X, t, k \mapsto \begin{cases} H(X, t) \\ P_0 P_2 - k \end{cases} \quad (7)$$

where H is defined as in equation (6) and let us state the two following remarks.

Remark 1 Let k_{sk} be the length $P_0 P_2$ read on the sketch X_{sk} . Let \mathcal{S} (respectively \mathcal{S}^R) be the homotopy path of H (resp. H^R) which belongs $(X_{sk}, 0)$ (resp. $(X_{sk}, 0, k_{sk})$). Denoting $p(\mathcal{S}^R)$ the projection of \mathcal{S}^R on its 10 first coordinates, we have $\mathcal{S} = p(\mathcal{S}^R)$.

Proof Let $f_k : \mathbb{R}^9 \rightarrow \mathbb{R}$ be the function defined for all $X \in \mathbb{R}^9$ that maps a figure X of $K_{3,3}$ to the length between its points P_0 and P_2 and consider the set $\mathcal{S}' = \{(X, t, f_k(X)) \in \mathbb{R}^{11} | (X, t) \in \mathcal{S}\}$. Since

- (i) \mathcal{S}' is connected (because f_k is continuous),
- (ii) $(X_{sk}, 0, k_{sk}) \in \mathcal{S}'$,
- (iii) $\forall (X, t, f_k(X)) \in \mathcal{S}', H^R(X, t, f_k(X)) = 0$,

we have $\mathcal{S}' \subseteq \mathcal{S}^R$ then $\mathcal{S} \subseteq p(\mathcal{S}^R)$.

Consider now the set $p(\mathcal{S}^R)$. Since it is connected, $(X_{sk}, 0) \in p(\mathcal{S}^R)$ and $\forall (X, t) \in p(\mathcal{S}^R), H(X, t) = 0$, we have $p(\mathcal{S}^R) \subseteq \mathcal{S}$. \square

Remark 2 The system of equations $(\mathcal{H}^R) : H^R(X, t, k) = 0$ can be rewritten as:

$$(\mathcal{H}^R) : \left\{ \begin{array}{l} \left. \begin{array}{l} P_0 P_1 - a_0(t) = 0 \\ P_1 P_2 - a_1(t) = 0 \\ P_0 P_2 - k = 0 \end{array} \right\} (\mathcal{B}^1) \\ \left. \begin{array}{l} P_2 P_3 - a_2(t) = 0 \\ P_0 P_3 - a_6(t) = 0 \end{array} \right\} (\mathcal{B}^2) \\ \dots \\ \left. \begin{array}{l} P_4 P_5 - a_4(t) = 0 \\ P_0 P_5 - a_5(t) = 0 \\ P_2 P_5 - a_8(t) = 0 \end{array} \right\} (\mathcal{B}^5) \end{array} \right\} (\mathcal{H}') \quad (8)$$

where $H'(X, t, k) = 0$ denotes the system of the 9 first equations involving unknowns (X, t, k) .

Considering k as a parameter, $(\mathcal{H}') : H'(X, t) = 0$ is triangular by blocks: each block (\mathcal{B}^i) of m_i equations involves exactly m_i unknowns that do not appear in blocks (\mathcal{B}^j) with $j < i$. Solving one block leads to construct one point.

We will say that (\mathcal{H}^R) is *quasi-triangularized* since except for the d last equations, here $d = 1$, \mathcal{H}^R is triangular by blocks. Given $G = C[X, A]$, finding d additional constraints in order to build H^R such that (\mathcal{H}^R) is quasi-triangularized can be achieved by combinatorial reasoning thanks to the reparameterization method (see [6]). Homotopy function H' corresponds to the homotopy function of a new GCSP, say $G' = C'[X, A']$, that differs from G only for d constraints. For the example $K_{3,3}$, G' is depicted in Fig. 7. Parameters of additional constraints of $C \setminus C'$ (here k) are called *driving parameters*. Finding the minimum number of additional constraints such that (\mathcal{H}^R) is quasi-triangularized is still an open problem.

2.4.1 Exploiting triangularization of (\mathcal{H}')

Knowing value for driving parameter k and interpolation parameter t , blocks of (\mathcal{H}') can be solved sequentially to obtain all its solutions. Here each block (\mathcal{B}^i) corresponds to a circle-line or a circle-circle intersection, and admits at most two solutions. Since (\mathcal{H}') is usually evaluated in a backtracking process, choosing a solution of (\mathcal{H}') amounts to choosing a branch in a binary tree where level n contains solutions for block (\mathcal{B}^n) . We can thus index different sequences of choices (*i.e.* different branches) and note them b_1, \dots, b_n .

For a given branch $b \in \{b_1, \dots, b_n\}$, we consider two functions: one for the sequential part (\mathcal{H}') of (\mathcal{H}^R) and another one for the d last equations. In our example, we define function P_b that maps values of t, k to the solution X of (\mathcal{H}') according to branch b . Next, function $f : (X, t) \mapsto P_2P_5 - a_8(t)$ is for the last equation. Then, solving (\mathcal{H}^R) for a given value of t amounts to seek values for $(k, b) \in \mathbb{R} \times \{b_1 \dots b_n\}$ such that

$$h_b(t, k) = f(P_b(t, k), t) = 0. \quad (9)$$

Obviously, functions P_b are not defined for any values of (t, k) since some intersections may be empty. The domain of definition of P_b will be denoted by $\mathcal{D}P_b$. It is in general neither open nor closed in the usual topology, but admits a boundary. Here, for instance, points (t, k) such that $k + a_1(t) = a_0(t)$ belongs to this boundary.

We will show in section 3 how to use constructive geometry to obtain a quasi-triangularization and continuous functions P_b that will be called *construction plan*. Their domains of definition will be studied and their boundaries characterized in terms of geometric configurations.

2.4.2 Paths of h_b

Considering h_b as homotopy functions, their homotopy paths can be characterized as one dimensional manifolds with boundaries. We will show in section 4 that the path \mathcal{S}^R of H^R to which belongs the sketch projects itself in a (finite) sequence of paths $(\mathcal{S}_i)_{1 \leq i \leq n}$ for n branches ((\mathcal{S}_i) for short), each \mathcal{S}_i being a path of h_{b_i} and having its boundaries in boundaries of $\mathcal{D}P_{b_i}$. Hence a point (t, k) belongs to the boundary of \mathcal{S}_i if and only if figure $P_{b_i}(t, k)$ presents a special geometric configuration.

Fig. 5 shows the projection on its two last coordinates t and k of a path \mathcal{S}^R of H^R for the GCSP $K_{3,3}$. This closed curve is a sequence of 4 paths \mathcal{S}_i , which boundaries are represented by black circles.

Considering remark 1, following the homotopy path \mathcal{S} of H containing the sketch can be achieved by following this sequence (\mathcal{S}_i) of paths, which requires at each prediction step and each Newton-Raphson iteration the inversion of a matrix of size $d \times d$, where d is the number of driving parameters (in the example $K_{3,3}$, $d = 1$).

However a major difficulty remains: path tracking processes are not robust near boundaries of $\mathcal{D}P_{b_i}$. First because h_b is not defined anywhere. When approaching a configuration of tangency

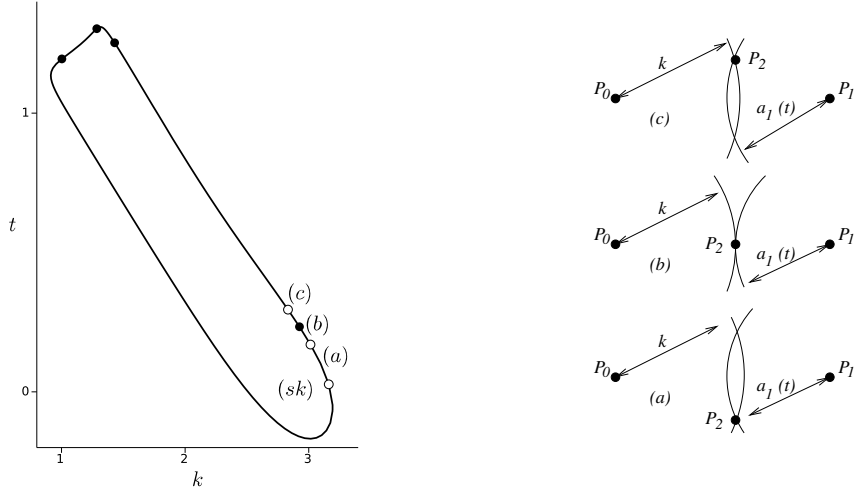


Figure 5: Left: the sequence of homotopy paths (delimited by black circles) of homotopy functions h_{b_i} corresponding to the projection of the path of H^R containing the sketch, that projects itself into (sk) . Figures corresponding to points (a), (b), (c) present geometric configurations (a), (b), (c) in right.

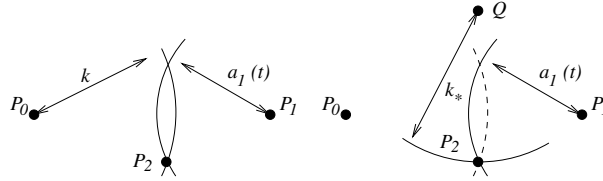


Figure 6: Left: the configuration (a) of right part of figure 5. Right: P_2 can be constructed by intersecting circle of center Q and ray k_* with circle of center P_1 and ray $a_1(t)$.

(circle-line or circle-circle intersection), the prediction can be made outside $\mathcal{D}P_{b_i}$, where there is no intersection. So, no correction is possible. Second, near tangency configurations h_b has very high derivatives values, leading quasi certainly the path tracking process to fail.

2.5 Changing reparameterization during the path-tracking process

We propose in section 5 a novel approach to overcome this problem that consists in changing the quasi-triangularization H^R during the path-tracking process when current point (k, t) is close to the boundary of h_b .

Suppose the configuration (a) of Fig. 5 is reached when tracking the path \mathcal{S}_1 of h_{b_1} from the sketch $(sk) = (k_{sk}, 0)$. It is close to the boundary configuration (b) what may lead the path tracking process to failure.

We change the quasi triangularization \mathcal{H}^R into \mathcal{H}^{R*} by replacing block \mathcal{B}^2 by block \mathcal{B}_*^2 defined as:

$$\left. \begin{aligned} P_1 P_2 - a_1(t) &= 0 \\ Q P_2 - k_* &= 0 \end{aligned} \right\} (\mathcal{B}_*^2)$$

Unknowns:
point P_0, \dots, P_5
Parameters:
length h_0, \dots, h_7, k
Constraints:
 $distance(P_0, P_1) = h_0$
 $distance(P_1, P_2) = h_1$
 $distance(P_0, P_2) = k$
...
 $distance(P_1, P_4) = h_7$

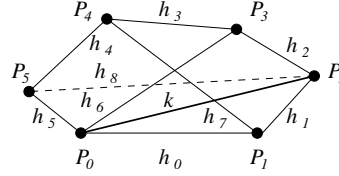


Figure 7: A symbolic statement (left part) and a dimensioned sketch (right part) of the GCSP *quasi* – $K_{3,3}$.

where Q is a new point which coordinates are fixed. \mathcal{H}^{R*} shares the same properties than \mathcal{H}^R hence it can be used to define a new construction plan and new homotopy functions h_b^* . The path \mathcal{S}^{R*} of H^{R*} containing the sketch projects itself on its two last coordinates k_* and t into a sequence (\mathcal{S}_i^*) of homotopy paths of h_b^* . Exploring path \mathcal{S} is then reduced to track paths (\mathcal{S}_i^*) from (a) instead of paths (\mathcal{S}_i) .

If Q is conveniently positioned (see for instance right part of Fig. 6) current point of tracked path is no more close to a boundary configuration and the tracking process can be continued.

The same method is applied each time a proximity of a boundary configuration is detected. It can lead to introduce new driving parameters when the block \mathcal{B}_i to be changed does not already involve any driving parameter.

3 Reparameterized construction plans

One way to get a quasi-triangularization is to perform a geometric solving by a ruler and compass construction. To go on with this geometric point of view, we will use the classical terms of geometric solving. So homotopy paths apply on a *construction plan* and blocks of equations are *instructions*.

3.1 Construction plan

Locus intersection method (LIM) is a simple method of geometric solving where the entities are sequentially constructed by intersecting loci. These intersections are placed in a list called construction plan. But some problems in 2D and most 3D problems can not be solved by this method. To remedy this, [6] proposes to delete some constraints and add other constraints to make the problem solvable by LIM. A numerical method then seeks to the added parameter values to satisfy the deleted constraints. How to choose the added and removed constraints is non-deterministic and described in [6]. This reparameterization process will not be discussed here.

Consider the GCSP depicted in Fig. 7 that has been obtained from the $K_{3,3}$ by substituting distance of parameter h_8 by distance between P_0 and P_2 of parameter k . We will call this GCSP *quasi* – $K_{3,3}$. This new distance k allows a simple ruler and compass construction for this problem. It consists of a sequence of intersections:

(I^0) P_0 lying on a line l_0 .

(I^1) Construct P_1 as intersection of l_0 and the circle of radius h_0 centered in P_0 .

Unknowns:
point P_1, \dots, P_5
Parameters:
point P_0 , line l_0
length $h_0, \dots, h_7, \mathbf{k}$
Terms:
 $P_1 = \text{InterCL}(P_0, h_0, l_0)$
 $P_2 = \text{InterCC}(P_0, \mathbf{k}, P_1, h_1)$
 $P_3 = \text{InterCC}(P_0, h_6, P_2, h_2)$
 $P_4 = \text{InterCC}(P_1, h_7, P_3, h_3)$
 $P_5 = \text{InterCC}(P_4, h_4, P_5, h_5)$

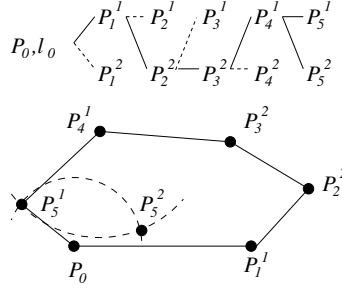


Figure 8: Left: A construction plan of solutions of the GCSP *quasi* – $K_{3,3}$. Right: a sub-tree of the associated interpretation tree, and the two figures $\{P_0^1, P_2^2, \dots, P_5^1\}$ and $\{P_0^1, P_2^2, \dots, P_5^2\}$ corresponding to the evaluation of the CP on the two branches in solid lines.

- (I^2) Construct P_2 as intersection of circles respectively centered in P_0 and P_1 of radius k and h_1 .
- ...
- (I^5) Construct P_4 as intersection of circles respectively centered in P_0 and P_4 of radius h_5 and h_4 .

Instruction I^0 fixes a reference to construct figures up to rigid motions. Each other instruction I^i constructs a geometric object by intersecting geometric loci defined by both parameters and previously constructed objects. Such a sequence of instructions is called a *Construction Plan* (CP) of the solutions of *quasi* – $K_{3,3}$.

More formally, a construction step I^i is a term $I^i[x, A]$ depending on a set of parameters A and constructing a geometric object x . For instance, the instruction $I^2[P_2, A^2]$ is written as $P_2 = \text{InterCC}(P_0, k_0, P_1, h_1)$ where $A^2 = \{P_0, k_0, P_1, h_1\}$ is the set of parameters, and P_2 the constructed object.

Definition 3 A *Construction Plan* P of objects X and parameters A with reference A^0 is a finite sequence $I = (I^i)_{i=1}^k$ of terms $I^i[x^i, A^i]$ such that

- (i) for all objects $x \in X$, either $x \in A^0$ or there is a term $I^i[x, A^i] \in I$ constructing x ,
- (ii) for all $I^i[x^i, A^i] \in I$, for all $a \in A^i$, either $a \in A$ or $a \in X$ is constructed by a term $I^k \in I$ with $k < j$.

We note $P = I[X \setminus A^0, A \cup A^0]$, or more simply $P = I[X, A]$.

If $G = C[X, A]$ is a GCSP, we say that $P = I[X, A]$ is a construction plan of its solutions if constructed figures fulfill all constraints of C . Fig. 8 shows a CP of solutions of *quasi* – $K_{3,3}$ with reference $\{P_0, l_0\}$.

3.1.1 Evaluation of a construction plan

For given values A of parameters and A^0 of reference, $P = I[X, A]$ can be evaluated, what means sequentially applying instructions of I . A multi-function, called *interpretation*, is associated to each instruction of P . For instance, *InterCC* is linked to a circle-circle intersection procedure.

This latter could yield one, two or no solutions (we assume that no solutions are returned if intersection is empty or if circles are the same).

Evaluating a CP leads to an interpretation tree. As for particular equation systems of section 2, each numerical solution is brought by a *branch* of that tree.

Right part of Fig. 8 shows a sub-tree of the interpretation tree associated to the CP presented in the left part, and the two figures resulting of its evaluation on the two branches in solid line.

Let $P = I[X, A]$ be a construction plan with reference A^0 and $I = \{I^1, \dots, I^n\}$. If I^i constructs unknown x^i from parameters A^i , we note $I_b^i(A^i)$ the function that maps to values A^i the result of the interpretation of I^i with choice b_i brought by b . Recall that $A^i \subset A \cup A^0 \cup \{x^1, \dots, x^{i-1}\}$ if $i > 1$, and $A^1 \subset A \cup A^0$. For the sake of simplicity, we note $I_b^1(A)$ for $I_b^1(A^1)$ and $I_b^i(A, x^1, \dots, x^{i-1})$ for $I_b^i(A^i)$.

We define the n functions P_b^i as

$$\begin{cases} P_b^1(A) = I_b^1(A) \\ P_b^i(A) = I_b^i(A, P_b^{i-1}(A)) \quad \text{if } 1 < i \leq n. \end{cases} \quad (10)$$

We note $P_b(A) = P_b^n(A)$ and we call it the numerical function associated to P on branch b . We note $P_b(A)$ the figure constructed by this function for values of parameters A .

It is straightforward that if each I_b^i is smooth (C^∞) in its variables A^i , P_b is smooth in its variables A , as a combination of smooth functions.

3.2 Reparameterization

3.2.1 Reparameterized construction plan

Let $G = C[X, A]$ be a GCSP and $G' = C'[X, A']$ be another one obtained by reparameterization, and P a construction plan for G' . $C' \setminus C$ is the set of d added constraints, depending on d driving parameters A^+ , and $C^- = C \setminus C' = \{c_0, \dots, c_{d-1}\}$ is the set of removed constraints.

We call *Reparameterized Construction Plan* (RCP) the pair $R = (P, C^-)$. If b is a branch of P , we will note $P_b(A^+, A)$ the numerical function associated to P on branch b to make appear the different roles played by driving parameters and other parameters.

For the following, it is worth mentioning that a RCP is not unique for a given GCSP.

By considering constraints of C^- with there numerical interpretations, we will note $c_i(X, A)$ the value of the numerical interpretation of constraint c_i .

We associate to $R = (P, C^-)$ the numerical functions

$$\begin{aligned} R_b : \mathbb{R}^d \times \mathbb{R}^l &\rightarrow \mathbb{R}^d \\ A^+, A &\mapsto \begin{pmatrix} c_0(P_b(A^+, A), A) \\ \dots \\ c_{d-1}(P_b(A^+, A), A) \end{pmatrix} \end{aligned} \quad (11)$$

where b is a branch of P , \mathbb{R}^l is the set of all possible values for A .

Since functions c_i are C^∞ , functions R_b are C^∞ .

3.2.2 Proof of the approach

Assuming that removed constraints can be evaluated for all X , which is the case for distance constraints, solving G for the values A_{so} is equivalent to finding couples (A_{so}^+, b) such that $R_b(A_{so}^+, A_{so}) = 0$.

Indeed, let X be a solution of G for A_{so} . It exists values A_{so}^+ that can be read on X s.t. X is a solution of G' for values of its parameters A_{so}^+, A_{so} . Hence it exists a branch b of P s.t. $P_b(A_{so}^+, A_{so}) = X$. Since X satisfies constraints of C^- , we have $R_b(A_{so}^+, A_{so}) = 0$.

If (A^+, b) is s.t. $R_b(A^+, A_{so}) = 0$, it satisfies $C \setminus C^-$ and C^- and $P_b(A^+, A_{so}) = X$ is a solution of G for values A_{so} .

3.3 Border of the domain of definition of a RCP

Be given a branch b , we are interested in characterizing the domain of definition \mathcal{DR}_b of function R_b and more precisely its border $\partial\mathcal{DR}_b$, that is a subset of $\mathbb{R}^d \times \mathbb{R}^l$. We will show that a point (A^+, A) belongs to this border if and only if the figure $P_b(A^+, A)$ presents one among a finite set of specific geometric configurations.

We recall that the border $\partial\mathcal{DR}_b$ of \mathcal{DR}_b is the subset of $\mathbb{R}^d \times \mathbb{R}^l$ such that for each $x \in \partial\mathcal{DR}_b$ and each $\epsilon \in \mathbb{R}^+$, a ball centered in x with ray ϵ contains both points of \mathcal{DR}_b and points of $(\mathcal{DR}_b)^c$ i.e. where R_b is not defined.

Assuming that functions c_0, \dots, c_{d-1} are defined for all possible figures X , \mathcal{DR}_b is exactly the domain of definition \mathcal{DP}_b of P_b , and $\partial\mathcal{DR}_b$ is exactly $\partial\mathcal{DP}_b$.

3.3.1 Border of \mathcal{DP}_b

Consider the definition of the numerical function P_b given in expression (10). Note \mathcal{DI}_b^i the domain of definition of numerical function associated to instruction I_b^i , $\partial\mathcal{DI}_b^i$ its border, and \mathcal{DP}_b^i the domain of definition of numerical function P_b^i , $\partial\mathcal{DP}_b^i$ its border.

Proposition 1 *Let P be the sequence of instruction $I = \{I^1, \dots, I^n\}$ and suppose that each instructions I^i involves at least one parameter that is not involved in other instruction. Then following sentences are equivalent.*

- (i) $(A^+, A) \in \partial\mathcal{DP}_b$
- (ii) \exists a smallest i s.t. $((A^+, A) \cup P_b^{i-1}(A^+, A)) \in \partial\mathcal{DI}_b^i$

We prove here that (ii) \Rightarrow (i) in the case where instruction I^i is the circle-circle intersection $x^i = \text{InterCC}(x^j, a^j, x^k, a^k)$. Suppose a^j is a parameter that is not involved in other instructions, and $((A^+, A) \cup P_b^{i-1}(A^+, A)) \in \partial\mathcal{DI}_b^i$. Let us enumerate the cases where (x^j, a^j, x^k, a^k) belongs to $\partial\mathcal{DI}_b^i$; all these cases correspond to specific geometric configurations.

- (1) the two circles are tangent because the triangular equality is satisfied,
- (2) the two circles are concentric with rays equal to zero,
- (3) the two circles are concentric with rays not equal to zero.

In cases (1) and (2), we have $(A^+, A) \in \mathcal{DP}_b^i$ since the two circles have an intersection. A ball centered in (A^+, A) of any ray $\epsilon \in \mathbb{R}^+$ contains values for which the two circles have no intersections: one can simply grow or decrease a^j by a sufficient small value. Then $(A^+, A) \in \partial\mathcal{DP}_b^i$.

In case (3) we consider that I_b^i is not defined in (x^j, a^j, x^k, a^k) , and as a consequence P_b^i is not defined in (A^+, A) , since intersections of the two circles are in infinite number. Suppose that one of the centers, say x_j , is defined by a previous instruction I^j , that involves the free parameter a_l . Since I_b^j is continuous, for any $\epsilon \in \mathbb{R}^+$, it is possible to find a point belonging to the ball centered in (A^+, A) with ray ϵ , resulting from a small variation of a_l , such that the two circles intersected in instruction I^i have a finite number of solutions. Hence $(A^+, A) \in \partial\mathcal{DP}_b^i$.

To finish the proof, we remark that $(A^+, A) \in \partial\mathcal{DP}_b$ if and only if it exists a smaller i s.t. $(A^+, A) \in \partial\mathcal{DP}_b^i$.

3.3.2 Border and boundary configurations

Proposition 1 states that (A^+, A) belongs to the border of \mathcal{DP}_b if and only if it exists i such that a constructed sub-figure $P_b^i(A^+, A)$ belongs to the border of the numerical interpretation of instruction I^i . One can now enumerate configurations corresponding to the border of I_b^i according to the symbol of I^i .

When I^i is a circle-circle intersection, these configurations are listed above. If a point (A^+, A) is such that figure $P_b^i(A^+, A)$ match cases (1) or (2), we have $(A^+, A) \in \mathcal{DP}_b$. If $P_b^i(A^+, A)$ match case (3), then $(A^+, A) \in (\mathcal{DP}_b)^c$.

When I^i is a line-circle intersection, figure $P_b^i(A^+, A)$ belongs to $\partial \mathcal{DI}_b^i$ if and only if

(4) the line is tangent to the circle

and $(A^+, A) \in \mathcal{DP}_b$.

We call *boundary* of \mathcal{DP}_b , and we note it $\bar{\mathcal{D}}\mathcal{DP}_b$ the set $\mathcal{DP}_b \cap \partial \mathcal{DP}_b$. We call specific configurations (1), (2) and (4) *boundary configurations*, and if a figure $P_b(A^+, A)$ presents one of these configurations for an instruction I^i , we say that it present a *boundary configuration*, and one can deduce that $(A^+, A) \in \bar{\mathcal{D}}\mathcal{DP}_b$.

4 Path tracking using RCP

Let G be the GCSP $C[X, A]$, H its associated homotopy function and $R = (P, C^-)$ a RCP involving d driving parameters. This RCP is a quasi-triangularization for the system of equations associated to G . We define the *R-extended homotopy function* $H^R : \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}^m \times \mathbb{R}^d$ which is obtained by appending d numerical functions corresponding to added constraints of R to components of H .

Let us recall a *R-extended homotopy function* for $K_{3,3}$:

$$H^R : \mathbb{R}^9 \times \mathbb{R} \times \mathbb{R} \rightarrow \begin{matrix} \mathbb{R}^{10} \\ \left\{ \begin{array}{l} P_0P_1 - a_0(t) \\ P_1P_2 - a_1(t) \\ P_0P_2 - k \\ P_2P_3 - a_2(t) \\ \dots \\ P_2P_5 - a_8(t) \end{array} \right. \end{matrix} \quad (12)$$

where a is an interpolation function from A_{sk} to A_{so} , and $P_0P_2 - k$ corresponds to the added constraint $k = \text{distance}(P_0, P_1)$.

Note $p : \mathbb{R}^{m+1+d} \rightarrow \mathbb{R}^{m+1}$ (respectively $p' : \mathbb{R}^{m+1+d} \rightarrow \mathbb{R}^{d+1}$) the projection of an object of \mathbb{R}^{m+1+d} on its $m+1$ first (resp. $d+1$ last) coordinates, and extend these notations to subsets of \mathbb{R}^{m+1+d} .

We first generalize the remark 1 of section 2:

Proposition 2 *If 0 is a regular value of H and numerical functions associated to added constraints are C^∞ over $\mathcal{DH} \times \mathbb{R}^d$ then 0 is a regular value of H^R . Moreover, if \mathcal{S} is an homotopy path of H , it exists a unique homotopy path \mathcal{S}^R of H^R such that $p(\mathcal{S}^R) = \mathcal{S}$, and \mathcal{S} and \mathcal{S}^R have the same topology.*

This proposition states that following a path \mathcal{S} of H can be achieved by following a path \mathcal{S}^R of H^R . The purpose of this section is to show that following \mathcal{S}^R , and hence \mathcal{S} , can be reduced to follow the set $p'(\mathcal{S}^R)$, which is a one dimensional set in a space of dimension $d+1$ instead of

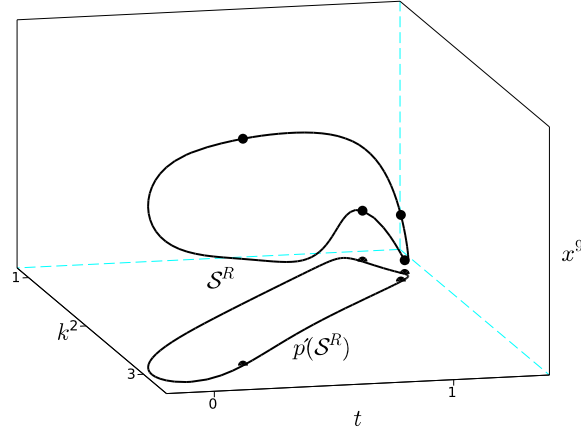


Figure 9: A projection of the path \mathcal{S}^R of the R -extended homotopy function for $K_{3,3}$ and its projection $p'(\mathcal{S}^R)$. Black points mark points of curves that present a boundary configuration.

$m + 1$. However, $p'(\mathcal{S}^R)$ is in general not a smooth manifold of dimension 1, but we will show that it can be decomposed into a finite sequence of such manifolds that are homotopy paths of functions build from R , and a theoretical algorithm that tracks this sequence will be given. Unfortunately the one who wants to implement this algorithm with a numerical path-tracking method will face difficulties, that are inherent to the multi-functional nature of the numerical interpretation of a construction plan.

From this proposition, two important remarks must be made.

Remark 3 *Be given a homotopy path \mathcal{S} of H , the path \mathcal{S}^R in proposition 2 is the subset of \mathbb{R}^{m+1+d} whose points are (X, t, A^+) where $(X, t) \in \mathcal{S}$ and A^+ are values of driving parameters that can be read on X .*

Remark 4 *Be given a homotopy path \mathcal{S}^R of H^R and a point $(X, t, A^+) \in \mathcal{S}^R$, it exists at least one branch b s.t $X = P_b(A^+, a(t))$.*

As a result, be given a point (t, A^+) of $p'(\mathcal{S}^R)$, it exists at least one branch b s.t $X = P_b(A^+, a(t))$ and $H^R(X, t, A^+) = H(X, t) = 0$.

The branch b in remark 4 is unique in most cases. However, if X presents a boundary configuration, there exists at least two such branches in our framework.

4.1 R -extended homotopy path

We give here some properties of a connected component \mathcal{S}^R of $(H^R)^{-1}(0)$. First, by proving proposition 2, we justify that \mathcal{S}^R is a one-dimensional smooth manifold. Then we will show that it contains a finite number n' of points (X, t, A^+) s.t. X presents a boundary configuration of R .

As a result, one could scan \mathcal{S}^R as a finite sequence $(\mathcal{S}_i^R)_{0 \leq i < n}$ of pieces of path, which two consecutive elements \mathcal{S}_i^R and \mathcal{S}_{i+1}^R share one point (X_i, t_i, A_i^+) s.t. X_i presents a boundary configuration of P .

If $n' = 0$, the sequence $(\mathcal{S}_i^R)_{0 \leq i < n}$ is reduced to one element $\mathcal{S}_0^R = \mathcal{S}^R$. Otherwise, if \mathcal{S}^R is diffeomorphic to a circle, $n = n'$ and each \mathcal{S}_i^R is diffeomorphic to $[0, 1]$. Else if \mathcal{S}^R is diffeomorphic

to $]0, 1[$ then $n = n' + 1$, \mathcal{S}_0^R and \mathcal{S}_{n-1}^R are diffeomorphic to $]0, 1[$ and \mathcal{S}_i^R , for $0 < i < n'$, are diffeomorphic to $[0, 1]$. Cases where \mathcal{S}^R is diffeomorphic to $[0, 1[$ or $[0, 1]$ are treated similarly.

Fig. 9 presents a projection on its three last coordinates x^9, t, k of the path \mathcal{S}^R (that is diffeomorphic to a circle) for the $K_{3,3}$. Black circles mark points that correspond to boundary configuration.

4.1.1 Proof of proposition 2

Let us write the i -th component of H as $H_i(X) - a_i(t)$, for $0 \leq i < m$. For the sake of the proof, we re-order components of H^R : its m first components are those of H and its d last correspond to added constraints and can be written $H_{m+i}(X) - k_i$ for $0 \leq i < d$ where k_0, \dots, k_{d-1} are the driving parameters of R . Hence, noting J_H the jacobian matrix of H , the jacobian matrix of H^R can be written:

$$J_{H^R} = \begin{pmatrix} (J_H) & (0) \\ (A) & (-Id) \end{pmatrix} \text{ where } A = \begin{pmatrix} \frac{\partial H_m}{\partial X} & 0 \\ \dots & 0 \\ \frac{\partial H_{m+d-1}}{\partial X} & 0 \end{pmatrix}$$

and is clearly of full rank if J_H is.

The set

$$\mathcal{S}^R = \{(X, t, H_m(X), \dots, H_{m+d-1}(X)) | (X, t) \in \mathcal{S}\}$$

is the unique homotopy path of H^R s.t. $p(\mathcal{S}^R) = \mathcal{S}$ and has the same topology than \mathcal{S} .

4.1.2 Boundary configurations

Proposition 3 \mathcal{S}^R contains only a finite number of points (X, t, A^+) s.t. X presents a boundary configuration of R .

When P involves only *InterCC* and *InterCL* instructions, boundary configuration are situations (1), (2) and (4) enumerated in subsection 3.3. Given a CP, it exists only finitely, say l , many different boundary configurations. Moreover, each of these can be characterized by a geometric constraint written as a polynomial equation $g_i = 0$.

Consider for instance the instruction $P_2 = \text{InterCC}(P_0, k, P_1, h_1)$ of the RCP for $K_{3,3}$. Configurations (1) or (2) hold for this instruction when circles of respective centers P_0, P_1 and rays k, h_1 are tangent. It is equivalent to say that P_0, P_1 and P_2 belongs to the same line, hence it can be translated in the equation $\det(\overrightarrow{P_0P_1}, \overrightarrow{P_0P_2}) = 0$.

Noting $Hg_i : \mathbb{R}^{m+1} \rightarrow \mathbb{R}^{m+1}$ the function obtained by appending g_i to H , a figure (X, t) presents a boundary configuration if and only if there is i s.t. (X, t) is a zero-value of Hg_i . We make the following assumption :

Assumption 1 $\forall 0 \leq i < l, 0$ is a regular value of Hg_i .

Next, by lemma on page 11 of [14], we know that roots of Hg_i are isolated points. Since Hg_i are polynomials, there are finitely many roots. \square

Cases when this assumption is false corresponds either to cases when parameters and interpolation function are non-generic, or to situations where g_i is a consequence of constraints of G , what can be very hard to detect.

4.2 Homotopy functions using RCP

Consider now \mathcal{S}^R as a finite sequence $(\mathcal{S}_i^R)_{0 \leq i < n}$. Tracking $p'(\mathcal{S}^R)$ amounts to follow a finite sequence $(\mathcal{S}_i)_{0 \leq i < n}$ where $\mathcal{S}_i = p'(\mathcal{S}_i^R)$ for $0 \leq i < n$. Each extremity of \mathcal{S}_i corresponds to a figure presenting a boundary configuration, and two consecutive elements \mathcal{S}_i and \mathcal{S}_{i+1} shares such a point.

Even if $p'(\mathcal{S}^R)$ is not always a manifold and can present for instance self-intersections, we will characterize each \mathcal{S}_i as a homotopy path of a homotopy function build from R . Fig. 9 presents the path \mathcal{S}^R for the GCSP $K_{3,3}$, and the projection $p'(\mathcal{S}^R)$ which each elements \mathcal{S}_i lie between two black circles.

4.2.1 Definition

We define the homotopy functions $h_b : \mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ as

$$h_b : t, A^+ \mapsto R_b(A^+, a(t)) \quad (13)$$

where b is a branch of P and R_b defined as in equation (11).

h_b is C^∞ over its domain of definition $\mathcal{D}h_b = \mathcal{D}R_b$. The border $\partial\mathcal{D}h_b$ will be studied. Since functions c_0, \dots, c_{d-1} are C^∞ for all $(X, t) \in \mathbb{R}^m \times \mathbb{R}$, $\mathcal{D}h_b$ (resp. $\partial\mathcal{D}h_b$) is exactly the domain of definition (resp. the border of the domain of definition) of the function $P_b \circ a : (t, A^+) \mapsto P_b(A^+, a(t))$.

4.2.2 Boundary configurations

We suppose that the interpolation function a is sufficiently generic and that proposition 1 can be extended to function $P_b \circ a$. Hence (t, A^+) belongs to $\partial\mathcal{D}h_b$ iff figure $P_b(A^+, a(t))$ presents a specific geometric configuration. More, supposing that instructions of P are of type *InterCC* or *InterCL*, $(t, A^+) \in \partial\mathcal{D}h_b$ iff the figure $P_b(A^+, a(t))$ presents one of the boundary configurations enumerated in 3.3.

In the latter case, it exists $b' \neq b$ s.t. (t, A^+) belongs to $\partial\mathcal{D}h_{b'}$ and $P_{b'}(A^+, a(t)) = P_b(A^+, a(t))$, and finally $h_{b'}(t, A^+) = h_b(t, A^+)$. Moreover, we suppose that b' is unique. It is not when X presents a boundary configuration for more than one instruction, but assumption 1 rules these cases out.

In other words, if $(t, A^+) \in \partial\mathcal{D}h_b$, then $(P_b(A^+, a(t)), t, A^+)$ is a solution of $H^R = 0$ with multiplicity two, and two branches b and b' allow to construct $P_b(A^+, a(t))$.

4.2.3 h_b vanishes on $p'(\mathcal{S}^R)$

Consider a point (t, A^+) of $p'(\mathcal{S}^R)$. It exists a branch b of P s.t. the point $(P_b(A^+, a(t)), t, A^+)$ belongs to \mathcal{S}^R . Hence $h_b(t, A^+) = 0$. More, one can state:

Proposition 4 *Let $(\mathcal{S}_i)_{0 \leq i < n}$ be the parsing of $p'(\mathcal{S}^R)$ defined above. For each $0 \leq i < n$, it exists b such that \mathcal{S}_i is a connected subset of $h_b^{-1}(0)$.*

Let $0 \leq i < n$. \mathcal{S}_i is connected since it is the projection of \mathcal{S}_i^R . Consider two points (X_1, t_1, A_1^+) and (X_2, t_2, A_2^+) of \mathcal{S}_i^R and suppose that X_1 does not present a boundary configuration. Hence it exists a unique branch b s.t. $X_1 = P_b(A_1^+, a(t_1))$ and $h_b(t_1, A_1^+) = 0$. Since the segment of path \mathcal{S}_i^R between (X_1, t_1, A_1^+) and (X_2, t_2, A_2^+) is a smooth deformation of (X_1, t_1, A_1^+) into (X_2, t_2, A_2^+) that does not pass through a boundary configuration, we have $X_2 = P_b(A_2^+, a(t_2))$ and $h_b(t_2, A_2^+) = 0$.

4.2.4 Homotopy paths of h_b

We note $h_b|_{\partial\mathcal{D}h_b}$ the restriction of h_b to the boundary of $\mathcal{D}h_b$, and we adapt to our notations a lemma that can be found on page 13 of [14]:

Lemma 1 *If 0 is a regular value both for h_b and for $h_b|_{\partial\mathcal{D}h_b}$, then connected components of $h_b^{-1}(0)$ are smooth manifolds of dimension 1 with boundary equal to their intersections with boundary of $\mathcal{D}h_b$.*

Assuming this regularity condition, components of $h_b^{-1}(0)$ are either diffeomorphic to circles or lines that do not have any boundary, or to segments $[0, 1[$ or $[0, 1]$ that have boundaries. In these last cases, a boundary point (t, A^+) is such that $P_b(A^+, a(t))$ presents a boundary configuration, and it exists a unique $b' \neq b$ with:

- (i) $(t, A^+) \in \partial\mathcal{D}h_{b'}$,
- (ii) $h_{b'}(t, A^+) = 0$,
- (iii) (t, A^+) belongs to a connected component of $h_{b'}^{-1}(0)$ that is diffeomorphic to either $[0, 1]$ or $[0, 1[$.

Hence, considering proposition 4, for each sub-set \mathcal{S}_i of $p'(\mathcal{S}^R)$, there is a branch b_i s.t. \mathcal{S}_i is a homotopy path of h_{b_i} with extremities in $\partial\mathcal{D}h_{b_i}$ and can be followed by a path-tracking method since it is a one dimensional manifold.

4.2.5 Regularity of h_b

It seems difficult to transfer properties of regularity of value 0 from H and Hg_i , for $0 \leq i < l$, to functions h_b and $h_b|_{\partial\mathcal{D}h_b}$, for all branches of P . Instead of showing that they hold, and then that hypothesis of lemma 1 are satisfied, we simply give some tracks to ensure that sets \mathcal{S}_i are sufficiently regular.

First, \mathcal{S}_i does not present any self-intersection. Consider \mathcal{S}_i^R and the branch b such that $X = P_b(A^+, a(t))$ for $(X, t, A^+) \in \mathcal{S}_i^R$. If (t, A^+) is a node of \mathcal{S}_i then (X, t, A^+) is a node of \mathcal{S}_i^R what contradicts the result of proposition 2. Remark that $p'(\mathcal{S})$ can present self-intersections, but in this case it is an intersection of sets \mathcal{S}_i and \mathcal{S}_j with $i \neq j$.

Then, $\mathcal{S}_i \setminus \partial\mathcal{S}_i$ is smooth: instead of a projection, it can be seen as the image of pieces of \mathcal{S} by the smooth function that maps to a figure values of driving parameters.

Finally, we assume that cases where the tangent of $\mathcal{S}_i \setminus \partial\mathcal{S}_i$ vanishes corresponds to non-generic cases.

4.3 Path tracking

Let us now expose the main result of this section. It states that an homotopy path \mathcal{S} of H can be deduced from a sequence of homotopy paths of homotopy functions h_b , build from a RCP.

We recall that following \mathcal{S} requires to invert the jacobian matrix of H which size is $m \times m$, while following homotopy paths of h_b leads to invert matrix of size $d \times d$ where $d \ll m$. The possible gain in computation time brought by this approach justifies our motivation.

Proposition 5 *Let \mathcal{S} be an homotopy path of H . It exists a finite sequence $\{(\mathcal{S}_0, b_0), \dots, (\mathcal{S}_{n-1}, b_{n-1})\}$ such that \mathcal{S}_i is an homotopy path of h_{b_i} with boundary $\partial\mathcal{S}_i \subset \partial\mathcal{D}h_{b_i}$ satisfying*

$$(i) \bigcup_{i=0}^{n-1} P_{b_i}(\mathcal{S}_i) = \mathcal{S},$$

(ii) $\partial\mathcal{S}_i \cap \partial\mathcal{S}_{i+1}$ contains exactly one point (t_{i+1}, A_{i+1}^+) if $i < n - 1$,

(iii) if \mathcal{S} is diffeomorphic to a circle, $\partial\mathcal{S}_{n-1} \cap \partial\mathcal{S}_0$ contains exactly one point (t_0, A_0^+) .

4.3.1 Proof of proposition 5

Let \mathcal{S} be an homotopy path of H , and 0 be a regular value of H . From proposition 2 it exists a unique path \mathcal{S}^R satisfying $p(\mathcal{S}^R) = \mathcal{S}$ and from proposition 3 the latter can be parsed in a finite sequence $(\mathcal{S}_i^R)_{0 \leq i < n}$. The projection $p'(\mathcal{S}^R)$ is then parsed in a finite sequence $(\mathcal{S}_i)_{0 \leq i < n}$ which each element \mathcal{S}_i is an homotopy path of a function h_{b_i} . More, \mathcal{S}^i has its boundaries in $\partial\mathcal{D}h_{b_i}$ and is a one dimensional manifold that can be followed by a path tracking method, thanks to lemma 1. The sequence $((\mathcal{S}_i, b_i))_{0 \leq i < n}$ clearly satisfies points (i), (ii) and (iii).

4.3.2 Theoretical path-tracking algorithm

The above proposition gives rise to a theoretical path-tracking algorithm that allows to explore a homotopy path $\mathcal{S} \in \mathbb{R}^{m+1}$ of H by following a sequence of path $(\mathcal{S}_i)_{0 \leq i < n}$ with $\mathcal{S}_i \in \mathbb{R}^{d+1}$.

Continuation solving of RCP

Input: GCSP $C[X, A]$, values A_{so} , sketch X_{sk} , RCP $R = (P, C^-)$ with driving parameters A^+ .

Output: list of solutions \mathcal{L}_{sol}

1. Read values A_{sk} and A_{sk}^+ from sketch X_{sk} , build an interpolation function a from A_{sk} to A_{so} .
2. Find the current branch b s.t. $P_b(A_{sk}^+, A_{sk}) = X_{sk}$ and build homotopy function $h_b(t, A^+) = R_b(A^+, a(t))$.
3. Let $t = 0$, and save the current branch as the branch of the sketch: $b_{sk} = b$.
4. Follow path of $h_b^{-1}(0)$ from (t, A_{sk}^+) until it loops on $(0, A_{sk}^+)$ with $b = b_{sk}$, while checking if :
 - $X = P_b(A^+, a(t))$ converges to a solution at infinity, then stop;
 - $X = P_b(A^+, a(t))$ converges to a non-derivable figure, then stop;
 - (t, A^+) crosses hyperplane $t = 1$, then append solution $X = P_b(A^+, a(t))$ to \mathcal{L}_{sol} ;
 - $(t, A^+) \in \partial\mathcal{D}h_b$, then find the unique branch b' s.t. $P_b(A^+, a(t)) = P_{b'}(A^+, a(t))$, do $b = b'$ and go to 4.

4.3.3 Implementation with a numerical method

Regardless to the fact that it fails if assumption 1 is not satisfied, the one who wants to implement the above algorithm with a numerical path tracking method will face two major difficulties.

The first one lies in the nature of numerical methods: calculated points are approximations of paths \mathcal{S}_i , and $(t, A^+) \in \partial\mathcal{D}h_{b_i}$ will never be exactly crossed. Hence a prediction-correction method will terminate with a predicted point outside of $\mathcal{D}h_{b_i}$ for which it is not possible to apply the correction method. A piecewise linear approximation will terminate when computing a simplex that has a vertex outside of $\mathcal{D}h_{b_i}$. Linear approximation of h_{b_i} can not be computed in this case.

This issue could be addressed by the use of commonly called watch-dogs to estimate to which geometric configurations correspond figures that belong to the set $\{P_{b_i}(A^+, a(t)) | (t, A^+) \in \partial\mathcal{D}h_{b_i}\}$ in order to identify the branch b_{i+1} such that $\partial\mathcal{D}h_{b_i} = \partial\mathcal{D}h_{b_{i+1}}$. Then, by appending an equation describing this geometric configuration to h_{b_i} , a point approximating $\mathcal{S}_i \cap \partial\mathcal{D}h_{b_i}$ can

be obtained by applying a numerical method, for instance Newton-Raphson iterations, to the obtained numerical function.

The second difficulty comes from the fact that more the current approximation (t, A^+) of \mathcal{S}_i is close to the boundary of $\mathcal{D}h_{b_i}$, higher are the values of the derivatives of h_{b_i} .

It can be easily remarked by considering a function that maps to a positive real number r the positive y -coordinate of the intersections of the unit circle with a circle centered in $(0, 2)$ of radius equal to r . Noting y this function, we have $y(r) = \frac{\sqrt{-r^4 + 10r^2 - 9}}{4}$ and $y'(r) = \frac{-4r^3 + 20r}{8\sqrt{-r^4 + 10r^2 - 9}}$. It is straightforward that $\mathcal{D}y = [1, 3]$ and that $\partial\mathcal{D}y = \{1, 3\}$ corresponds to configurations where the circles are tangent; we have $\lim_{r \rightarrow \partial\mathcal{D}y} y'(r) = +\infty$. Notice that it is not due to the chosen system of coordinates.

It can give rise either to a growing inexactitude when approaching a boundary configuration if the path is tracked with a constant step. Otherwise, if the step is adapted during the tracking, the former can be decreased infinitely.

5 On-the-fly change of RCP

We propose here a new approach to explore entirely a homotopy path \mathcal{S} of H constructed from a GCSP G without tracking it, that uses reparameterization to decrease the cost of path tracking, but overcomes the difficulties pointed out in section 4.

As stated before, a reparameterization is not unique. Moreover, when a figure presents a boundary configuration for a given reparameterization, the latter can be slightly modified in order to avoid this situation. Subsection 5.1 shows how to find this new reparameterization. An algorithm will be proposed in subsection 5.2 which basic idea is to change on-the-fly, during the path tracking process, the reparameterization used to define the tracked path when it is detected to be close to a border configuration. The implementation of this algorithm requires to specify the notion of closeness to a boundary configuration. It is done in subsection 5.3 by defining a dedicated geometric criteria.

In this section, $G = C[X, A]$ is a 2D GCSP. For the sake of simplicity it involves only constraints of distances. H is its associated homotopy function.

5.1 Different RCP

For a given problem G , we put forward two points about considering different RCP which our method will take advantage. Let \mathcal{S} be a homotopy path of H .

The following assertion that comes directly from proposition 2 and states that different RCP lead to follow the same path.

Proposition 6 *Let R_1 and R_2 be two RCP of G , then it exists a unique homotopy path \mathcal{S}^{R_1} (resp. \mathcal{S}^{R_2}) of the R_1 -extended (resp. R_2 -extended) homotopy function such that $p(\mathcal{S}^{R_1}) = p(\mathcal{S}^{R_2}) = \mathcal{S}$.*

As a result, applying the procedure **Continuation solving of RCP** described in section 4 with different RCP of G leads to compute projections of the same path \mathcal{S} .

But it is important to notice that different RCP have different boundary configurations.

Proposition 7 *Let R_1 be a RCP, and $(X, t) \in \mathcal{S}$ s.t. X presents a boundary configuration of R_1 . It exists a RCP R_2 s.t. X does not present a border configuration of R_2 .*

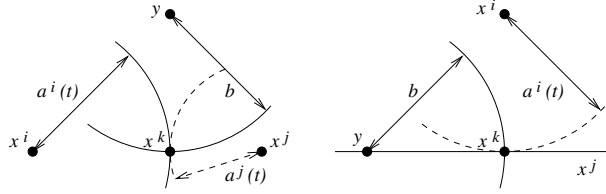


Figure 10: Left: the sub-figure $\{x^i, x^j, x^k\}$ presents a boundary configuration of instruction $x^k = InterCC(x^i, a^i(t), x^j, a^j(t))$, but not for $x^k = InterCC(x^i, a^i(t), y, b)$. Right: the same with instructions $x^k = InterCL(x^i, a^i(t), x^j)$ and $x^k = InterCL(y, b, x^j)$.

We prove this proposition by showing how to construct R_2 . Let $R_1 = (P, C^-)$, with d driving parameters A^+ , and $P = I[X \setminus A^0, A \cup A^0 \cup A^+]$ (A^0 are the unknowns used to build the reference). Since G involves only distance constraints, instructions of P have type *InterCC* or *InterCL*.

If X presents a boundary configuration, it exists a smaller k s.t. X presents a boundary configuration of instruction I^k of P . Consider first that I^k does not involve any driving parameter.

Let I^k be $x^k = InterCC(x^i, a^i(t), x^j, a^j(t))$; hence in the figure X , circles of centers x^i and x^j with radii $a^i(t)$ and $a^j(t)$ are either tangent (case a), or concentric with vanishing radii (case b).

In case a, consider a new point $y \neq x^i, x^j$. Then circles of centers x_i and y , with radii $a^i(t)$ and $\|yx^k\|_2$ are not tangent, and intersect in x^k , as it is illustrated in the left part of Fig. 10. One can define the instruction $I^{k'}$ as $x^k = InterCC(x^i, a^i(t), y, b)$ and the construction plan $P' = I'[X \setminus A^0, A \cup A^0 \cup \{y\} \cup A^+ \cup \{b\}]$ obtained from I by replacing I^k with $I^{k'}$. Then $R_2 = [P', C^{-'}]$ where $C^{-'} = C^- \cup \{a_j = distance(x_i, x_j)\}$ is a RCP of G with $d + 1$ driving parameters $A^+ \cup \{b\}$ for which X does not present a boundary configuration.

In case b, it is necessary to introduce two new points y and z and two new driving parameters to construct the instruction $I^{k'}$.

The construction of $I^{k'}$ when I^k is an *InterCL* instruction is illustrated in right part of Fig. 10.

Consider now that I^k involves a driving parameter r , say $x^k = InterCC(x^i, a^i(t), x^j, r)$, and that circles of centers x^i and x^j with radii $a^i(t)$ and r are tangent (the other cases are similar). y being a new point defined as in case a, we define $I^{k'}$ as $x^k = InterCC(x^i, a^i(t), y, b)$ with b a new driving parameter, P' as previously, and R_2 as $[P', C^{-'}]$. Hence R_2 is a RCP of G with d driving parameters $(A^+ \setminus \{r\}) \cup \{b\}$ for which X does not present a boundary configuration.

5.2 Changing RCP during the path-tracking process

We rely on propositions 6 and 7 to design a new algorithm to compute the path \mathcal{S} of H containing the sketch using reparameterization to decrease the cost of the tracking process. To each instruction I^k of the construction plan is associated a geometric criterion γ^k which role is to decide when a figure X corresponding to a point of the tracked path is close to a boundary configuration. When this situation arises, the RCP used to define the tracked path is modified as shown in the proof of proposition 7. This new RCP defines a new path that is followed from the point corresponding to the last computed figure, until it is close to a new boundary configuration (and the RCP is one more time changed) or it reaches a termination condition.

Applying this method leads to track a sequence of pairs $(\mathcal{S}_i, R_i)_{i \in N \subseteq \mathbb{N}}$ where N is an internal of \mathbb{N} and for each i , \mathcal{S}_i is a homotopy path of the homotopy function using RCP R_i . In 5.2.3 we

will show that if the algorithm terminates, *i.e.* N is bounded, the set of computed figures is \mathcal{S} . Subsection 5.2.4 justifies termination.

5.2.1 Main algorithm

To each instruction I^k , for $1 \leq k \leq n$, we associate the real univariate C^∞ function γ^k , that maps to a figure X an element of $[0, 1]$. Be given $\alpha \in]0, 1[$, we consider that a figure X is close to a boundary configuration when $\min_k \gamma^k(X) \leq \alpha$. Functions γ^k are defined in subsection 5.3.

Continuation solving of RCP with adaptive CP

Input: $G = C[X, A]$, values A_{so} , sketch X_{sk} , RCP $R = (P, C^-)$ with driving parameters A^+ , $\alpha \in]0, 1[$.

Output: list of solutions \mathcal{L}_{sol}

1. Read values A_{sk} from sketch X_{sk} , build an interpolation function a from A_{sk} to A_{so} ; let $t = 0$.
2. Let $X_{cur} = X_{sk}$.
3. Perform procedure **Initialize RCP** with R and X as input values.
4. Read values A_{cur}^+ on X_{cur} , find the current branch b s.t. $P_b(a(t), A_{cur}^+) = X_{cur}$ and build homotopy function $h_b(A^+, t) = R_b(A^+, a(t))$.
5. Follow path of $h_b^{-1}(0)$ from (A_{cur}^+, t) until current figure $X = P_b(a(t), A^+)$ loops on X_{sk} while checking if :
 - X converges to a solution at infinity, then stop;
 - X converges to a non-derivable figure, then stop;
 - (A^+, t) crosses \mathcal{P}_1 , then append solution X to \mathcal{L}_{sol} .
 - It exists $I^k[X^k, A^k]$ in P s.t. $\gamma^k(X) < \alpha$, then perform procedure **Change RCP** with R , X , k and α as input values, let $X_{cur} = X$ and go to 4.

Steps 1 to 3 are initialization steps. The procedure **Initialize RCP** is rather technical, and is informally described below; its necessity will emerge from the procedure **Change RCP**.

Step 4 is performed each time the RCP is changed. It consists in reading on X_{cur} values of new driving parameters A_{cur}^+ and branch b . Suppose the instruction introduced when changing the RCP is $x^k = InterCC(x^i, a^i(t), y, s)$, then s is set to the value of the distance between x^k and y . Finding b is simply performed by evaluating P and choosing at each step the result among that two corresponds to X_{cur} . The cost of this step is linear in the number of instructions of P .

A technical point lies in step 5: how to ensure that there is no track-back on \mathcal{S} between \mathcal{S}_i and \mathcal{S}_{i+1} . It is addressed by checking the progression on \mathcal{S} .

5.2.2 Procedure Change RCP

To change the RCP in the procedure **Change RCP** we consider the instruction I^k which current figure X reaches the boundary of the domain of definition. That is we have $\min_i \gamma^i(X) = \gamma^k(X) < \alpha$.

This procedure introduces new driving parameters, and we need to distinguish among the set of driving parameter those that were involved in the RCP given as input of the main procedure. They will be called *original*.

Suppose I^k is $x^k = InterCC(x^i, a^i, x^j, a^j)$, and switch on the number of driving parameters in $\{a^i, a^j\}$.

Case 0 One couple among $(a^i, x^i), (a^j, x^j)$, say (a^j, x^j) , is chosen to construct $I^{k'}$ and the constraint $a^j = \text{distance}(x^k, x^j)$ is merged to C^- . Let $I^{k'}$ be $x^k = \text{InterCC}(x^i, a^i(t), y, b)$ where y is a new point which placement is described in 5.3 and b a new parameter. R is modified as follows: $A^0 = A^0 \cup \{y\}$, $A^+ = A^+ \cup \{b\}$, $I^k = I^{k'}$ and $C^{-'} = C^- \cup \{a_j = \text{distance}(x_i, x_j)\}$, and the procedure is stopped.

Case 1 Let a^j be the driving parameter: either it has been introduced at the time of a previous call by case 0, then perform step a), or it is an original driving parameter then perform step b).

- a) Let $I^{k'}$ be the instruction $x^k = \text{interCC}(x^i, a^i, x^{j'}, a^{j'})$ that was replaced by I^k . If $\gamma^{k'}(X) > \alpha$, $I^{k'}$ is restored, A^+ is set as $A^+ \setminus \{a^j\}$, A^0 as $A^0 \setminus \{x^j\}$, C^- as $C^- \setminus \{a^{j'} = \text{distance}(x^k, x^{j'})\}$ and the procedure is stopped. If $\gamma^{k'}(X) \leq \alpha$, perform step b).
- b) We assume $x^j \in A^0$. Since it is a reference, it can be moved in combination to a^j without modifying X . Then x^j is moved in such a way that $\gamma^k(X) \geq \alpha$ and a^j is read on $X \cup \{x^j\}$.

Case 2 This situation never happen by construction.

This procedure raises three observations. First it is assumed in step 1b) that if a^j is a driving parameter in $x^k = \text{InterCC}(x^i, a^i, x^j, a^j)$ then $x^j \in A^0$, what is not true for original driving parameters. Actually, the role of the procedure **Initialize RCP** is to introduce a new point y , with $y = x^j$, for each driving parameter of the original RCP, and to appropriately modify the instruction involving a^j .

The second point is that procedure **Change RCP** increases the number of driving parameters each time it enters in case 0, and decreases it only if it performs step 1a) when the original instruction can be restored. To decrease more systematically the number of driving parameter, an initial step is perform when entering procedure **Change RCP**. It consists in applying **Remove driving parameters** with R, X, k and α as inputs. The procedure **Remove driving parameters** accepts as inputs a *RCP*, the current figure X and the index k of the instruction to be modified. For each instruction I^j with $j \neq k$ involving a non-original driving parameter, it restores the corresponding original instruction $I^{j'}$ if $\gamma^{j'}(X) > \alpha$.

The third observation is that it is straightforward to transpose this procedure to treat *interCL* instructions.

5.2.3 Correctness

The goal of this section is to show that when applying procedure **Continuation solving of RCP with adaptive CP**, the projection in the figures space of the obtained sequence of paths is \mathcal{S} , the path of H containing the sketch X_{sk} . Let $(\mathcal{S}_i, R_i)_{i \in N \subseteq \mathbb{N}}$ be the sequence of couples (paths, RCP) computed when applying procedure **Continuation solving of RCP with adaptive CP**, and suppose N is bounded. For $i \in N$, $H^{R_i} : \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^{d_i} \rightarrow \mathbb{R}^m \times \mathbb{R}^{d_i}$ is the R_i -extended homotopy function and $p'_i : \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^{d_i} \rightarrow \mathbb{R} \times \mathbb{R}^{d_i}$ the projection of a point (or a subset) of $\mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^{d_i}$ on its $d_i + 1$ last coordinates. Recall that p is the projection on the $m + 1$ first coordinates.

Each path \mathcal{S}_i is the projection by p'_i of a 1-dimensional manifold \mathcal{S}^{R_i} that is a subset of a homotopy path of H^{R_i} . Consider $(\mathcal{S}^{R_i})_{i \in N}$. We will show that if \mathcal{S} is diffeomorphic to a circle, $\bigcup_{i \in N} p(\mathcal{S}^{R_i}) = \mathcal{S}$.

When performing for the first time the step 4 one have $X_{cur} = X_{sk}$, hence noting P the construction plan of R_0 , b is s.t. $P_b(A_{sk}^+, 0) = X_{sk}$, then $(X_{sk}, 0, A_{sk}^+) \in \mathcal{S}^{R_0}$ and we deduce that $p(\mathcal{S}^{R_0}) \subseteq \mathcal{S}$.

If $N = \{0\}$, *i.e.* procedure terminates without entering in **Change** RCP because path overlaps, then \mathcal{S}^{R_0} is diffeomorphic to a circle and $p(\mathcal{S}^{R_0}) = \mathcal{S}$.

Otherwise, naming X_{cur} the figure corresponding to the last point (t_i, A^i) of \mathcal{S}_i computed before entering the procedure **change** RCP for the $i+1$ -th time, P the construction plan of R_{i+1} and A^+ its driving parameters, then when performing for the $i+1$ -th time the step 4, b is chosen s.t. $P_b(A_{sk}^+, t_i) = X_{cur}$. Then $p(\mathcal{S}^{R_i})$ and $p(\mathcal{S}^{R_{i+1}})$ are connected subsets of a homotopy path of H . If there is no backtrack, $p(\mathcal{S}^{R_i}), p(\mathcal{S}^{R_{i+1}})$ share exactly one point if N contains more than two elements³. Considering $p(\mathcal{S}^{R_i}) \subseteq \mathcal{S}$ as a recurrence hypothesis, we have $p(\mathcal{S}^{R_{i+1}}) \subseteq \mathcal{S}$.

If the figure corresponding to the last extremity of the tracked path is the sketch, then $\bigcup_{i \in N} p(\mathcal{S}^{R_i})$ is diffeomorphic to a circle, what finishes the proof.

When \mathcal{S} is diffeomorphic to an interval the procedure has to be performed two times, to explore \mathcal{S} from the sketch in both directions.

5.2.4 Termination

Here we consider a path tracker as an abstract object that can be iterated. It either successes and progresses along a path, returning a new point and stops when a termination condition is reached or fails only if current figure X reaches a boundary configuration, *i.e.* $\min_i \gamma^i(X) = \gamma^k(X) \leq \alpha$.

Applying our algorithm yields to perform a sequence s of steps, each being of one of the types:

- (a) successful iteration of the path-tracker,
- (b) failing iteration of the path-tracker,
- (c) call to procedure **Change** RCP.

In a step of type (a), a termination condition is eventually reached and the algorithm stops without error.

We build our discussion on a disjunction on infinite sequences s knowing that a step (a) is followed either by a step (a) or a step (b), a step (b) is followed by a step (c), and a step (c) is followed either by a step (a) or (b).

Suppose s contains an infinite number of consecutive (b) and (c) steps: the procedure **change** RCP is inefficient. A straightforward necessary condition that avoid this situation is the following. If X is the last figure computed, R_i the RCP which X satisfies $\gamma^k(X) \leq \alpha$ and R_{i+1} the new RCP constructed by **change** RCP then $\gamma^{k'}(X) > \alpha$ if $\gamma^{k'}$ is the geometric criterion associated to the new instruction $I^{k'}$ in R_{i+1} . In subsection 5.3 are given a definition of γ^k and a geometric placement of new points that ensure this necessary condition. Assuming this hypothesis, the RCP is invariant after a finite number of consecutive steps (b) and (c), and the current figure X satisfies $\min_i \gamma^i(X) > \alpha$. Hence the path-tracker should progress along the path.

Consider s contains an infinite number of consecutive steps (a). Recall that the tracked path \mathcal{S}^i is the projection of the path \mathcal{S}^{R_i} that is diffeomorphic to a circle and has a finite arc-length. We have made the assumption that each iteration of the path-tracker progresses along the path. Measuring this progression with the arc-length of \mathcal{S}^i between two consecutive points, we suppose that it exists $\delta > 0$ s.t. at each iteration, this progression is greater that δ . Hence a termination condition is reached after a finite number of steps (a).

³otherwise the other extremity of \mathcal{S}^{R_1} is the sketch and paths \mathcal{S}^{R_0} and \mathcal{S}^{R_1} share exactly two points

Suppose s is an infinite sequence of patterns a...abc...bc, where each pattern is composed by a finite number of steps (a) followed by a finite number of steps (b) and (c). Each iteration of the pattern yields a progression along a path \mathcal{S}^i , hence it yields a progression along \mathcal{S} . Nevertheless, it seems not clear that it exists $\delta > 0$ s.t. each progression along \mathcal{S} is greater than δ . Think for instance to a sequence of $(\mathcal{S}^i)_i$ which arc-lengths grow and are unbounded with i . Remark that for a given GCSP, it exists only a finite number of RCP, and a finite number of R -extended homotopy functions, each admitting a unique homotopy path \mathcal{S}^R that contains the sketch. The geometry of \mathcal{S}^R is linked to the placement of points added by the procedure **Change RCP**. But if we consider that figures X and added points stays in a compact set, \mathcal{S}^R belongs to a compact and has a bounded arc-length. Finally, considering that it exists a longest path \mathcal{S}^R , we claim that it exists $\delta > 0$ that is smaller than the progression along \mathcal{S} at each iteration. Hence a termination condition is reached after performing a finite number of patterns a...abc...bc.

5.3 Geometric criteria

The main procedure given in 5.2.1 and procedure **change RCP** need geometric criteria to be implemented. First, for each instruction I^k , with respect to its symbol, we associate in 5.3.1 a real positive function γ^k that maps to a figure an estimation of how close is the figure to a boundary configuration. Then **change RCP**, when it performs steps 1. (resp. 2b) introduces a new point (resp. moves a reference point). A way of placing this new point will be given in 5.3.2.

Finally, we will show in 5.3.3 that the placement of the new point is coherent with the stopping criterion, *i.e.* if I^k is such that $\gamma^k(X) \leq \alpha$ and $I^{k'}$ is the new instruction then $\gamma^{k'}(X) > \alpha$.

5.3.1 Stopping criteria

If I^k is the instruction $x^k = InterCC(x^i, a^i, x^j, a^j)$, we define

$$\gamma^k(X) = \begin{cases} c/a^i & \text{if } a^j \leq a^i \\ c/a^j & \text{otherwise} \end{cases} \quad (14)$$

where c is the distance between x^k and the line $x^i x^j$.

If I^k is the instruction $x^k = InterCL(x^i, a^i, x^j)$, we define

$$\gamma^k(X) = c/a^i \quad (15)$$

where c is the distance between x^k and the projection p of x^k on line x^j .

5.3.2 Placement of new points

In steps 1 and 2.b of procedure **change RCP**, one needs to place (*i.e.* to find coordinates of) the point y .

If I^k is the instruction $x^k = InterCC(x^i, a^i, x^j, a^j)$, let p be the projection of x^k on the line $x^i x^j$, and $\overrightarrow{x^k p}/x^k p$ the unit vector which direction is perpendicular to the line $x^i x^j$. We use the placement of y :

$$y = x^k + \max(a^i, a^j) \frac{\overrightarrow{x^k p}}{x^k p} \quad (16)$$

If I^k is the instruction $x^k = InterCL(x^i, a^i, x^j)$, we propose to place y on the line x^j , at a distance a^i of x^k :

$$y = x^k + a^i \frac{\overrightarrow{x^k p}}{x^k p} \quad (17)$$

where p is the projection of x^i on line x^j .

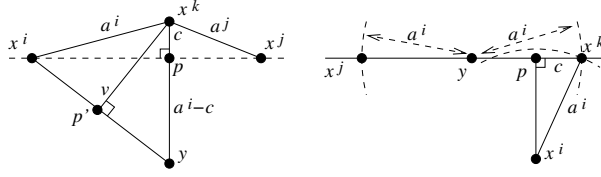


Figure 11: Left (resp. right): placement of the new point y for the instruction $x^k = \text{InterCC}(x^i, a^i, x^j, a^j)$ (resp. $x^k = \text{InterCL}(x^i, a^i, x^j)$) when $a^j \leq a^i$.

5.3.3 Coherence of criteria

Let $\alpha \leq 1$ and suppose $\gamma^k(X)$ is such that $\gamma^k(X) < \alpha$. Suppose that $I^{k'}$ is the instruction obtained after applying the procedure `change RCP`, with placement of y as described above. If $\alpha \leq 1/2$, then $\gamma^{k'}(X) > \alpha$.

It is straightforward when I^k is an *InterCL* instruction; see right part of Fig. 11. When I^k is the instruction $x^k = \text{InterCC}(x^i, a^i, x^j, a^j)$, with $a^j \leq a^i$, it is easy to show that $\frac{v^2}{(a^i)^2} = \frac{1}{2} + \frac{c}{2a^i}$ and to deduce the result. It is illustrated by left part of Fig. 11.

6 Implementation and results

The procedure `Continuation solving of RCP with adaptive CP` described in previous section has been implemented in `C++`, giving rise to a program that accepts a GCSP G , an initial RCP R of G , a sketch X_{sk} and values A_{so} . Numerical path-tracking process is achieved by a simple prediction-correction method with fixed step. It is described in subsection 6.2.

In subsection 6.3, we present numerical results about the application of our method on four point-point distances problems, one in 2D and three in 3D. It confirms that the approach of [9] that consists in following the path \mathcal{S} of H which belongs the sketch is fast and provides several solutions (sometimes all of them) of problems that resist to divide and conquer methods and are too large to be solved by classical homotopy or branch and bound approach. The method presented here uses parameterized construction plans and brings an important speed-up of this approach while staying robust.

We first give details about its generalization to 3D cases.

6.1 Generalization to 3D GCSP

3D GCSP with point-point distance constraints fit well to the solving method depicted in this paper: results of [9] as well as reparameterization approach stay valid. One important issue when solving a 3D GCSP is to decide if it is generically well constrained since there is not known combinatorial or structural characterization of the latter property. It is assumed to hold for GCSP considered here.

6.1.1 RCP in a 3D geometric universe

A reference consists in a point P_0 , a line l_0 and a plane pl_0 . A CP of a point-point distance GCSP in 3D has the structure:

$$(I^1) P_1 = \text{InterSL}(P_0, h_0, l_0)$$

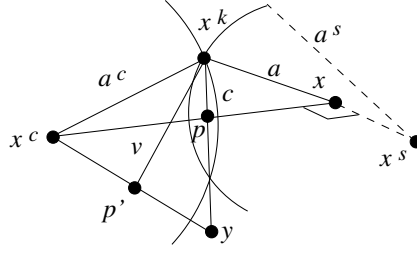


Figure 12: Placement of point y in cases of *InterSC* instructions. Dashed lines does not belong to the plane of circle centered in x^c .

$$(I^2) P_2 = \text{InterSSP}(P_0, h_1, P_1, h_2, pl_0)$$

$$(I^3) P_3 = \text{InterSSS}(P_0, h_3, P_1, h_4, P_2, h_5)$$

$$(I^4) \dots$$

where *InterSL* holds for a sphere-line intersection, *InterSSP* for the intersection of two spheres and one plane, and *InterSSS* for a three spheres intersection.

It is simpler to see a three spheres (resp. two spheres and one plane) intersection as the intersection of two spheres, which is a circle of non strictly-positive ray, and then the intersection of a circle and a sphere (resp. a plane). We will write *InterSS* for two spheres intersections, and *InterSC* (resp. *InterPC*) for sphere-circle (resp. plane-circle) intersections.

6.1.2 Border and boundary configurations

Possible border and boundary configurations of 3D parameterized construction plans can be enumerated and are similar to those encountered in a 2D context.

InterSS and *InterSL* are straightforward generalizations of *InterCC* and *InterCL* instructions.

InterSC instructions can be simplified in *InterCC* instructions by considering the intersection of the sphere and the plane which belongs the circle. *InterPC* is seen as a *InterCL* instruction provided that the plane is not parallel or equal to the plane which belongs the circle.

6.1.3 Stopping criteria and placement of new points

To generalize the algorithm in a 3D universe, one need to define stopping criteria and placement of new points for 3D instructions.

For *InterSS* and *InterSL*, instructions, we use the direct 3D generalization of stopping criteria and placement of new points of *InterCC* and *InterCL* instructions. Consider for instance the instruction $x^k = \text{InterSS}(x^i, a^i, x^j, a^j)$. Then the stopping criterion $\gamma^k(X)$ associated to it is $c/\max(a^i, a^j)$ where c is the distance of x^k to the line $x^i x^j$. Suppose $a^j < a^i$. When swapping this instruction with $x^k = \text{InterSS}(x^i, a^i, y, b)$, one can choose $b = a^i$ and $y = x^k + a^i \overrightarrow{(x^k p / x^k p)}$ where p is the projection of x^k on line $x^i x^j$.

We propose the following definition for the stopping criterion associated to an *InterSC* instruction. Let $x^k = \text{InterSC}(x^s, a^s, x^c, a^c)$, x be the projection of x^s on the plane pl which belongs the circle (x^c, a^c) , and c be the distance from point x^k to line $x^c x$. Then $\gamma^k(X) = c/\max(a^s, a^c)$.

When entering the procedure `change RCP`, if $\gamma^k(X) < \alpha$, the instruction $x^k = \text{InterSC}(x^s, a^s, x^c, a^c)$ is changed in $x^k = \text{InterSC}(y, b, x^c, a^c)$ where y is $y = x^k + a^c \frac{\overrightarrow{x^k p}}{x^k p}$ where p is the projection of x^k on line $x^c x$ and $b = a^c$. It is easy to show that this criterion is coherent in the sense of 5.3.3. The placement of y is depicted on Fig. 12.

6.2 Implementation

6.2.1 Path tracking

A path \mathcal{S} is followed from an initial point (t_0, d_0^+) by a classical prediction-correction method: prediction is performed by an Euler predictor with constant step $\delta \in \mathbb{R}$ and correction by Newton-Raphson iterations. Notice that it is a naive path tracking algorithm that does not ensure that there is no jump between two close paths. Clever approaches which merely rely on interval arithmetic (see [10, 4, 13]) have to be used to guarantee a path tracking algorithm. Here we just suppose that δ is small enough to follow considered curve. In the examples considered, such jumps never occurred.

The jacobian matrix $J(t_0, d_0^+)$ is numerically computed with the finite difference method, and the 1-dimensional kernel T of the linear application $J(t_0, d_0^+)$ is obtained with basics linear algebra tools. A prediction $(t'_0, d_0^{+'})$ is then computed in the direction of T , with prediction step δ . This prediction is then corrected with Newton's method in $(t_0^*, d_0^{+*}) \in \mathcal{S}$ such that $(t_0^*, d_0^{+*}) \in T_p$, where T_p is the hyperplane perpendicular to T . When (t_0, d_0^+) and the prediction $(t'_0, d_0^{+'})$ are such that $(t_0 - 1) \cdot (t'_0 - 1) < 0$ (*i.e.* hyperplane \mathcal{P}_1 lies between these two points), a new prediction $(t''_0, d_0^{+''})$ such that $t''_0 = 1$, is performed, and $(t''_0, d_0^{+''})$ is corrected in $(1, d_0^{+*})$ to precise the solution.

Prediction is made along line T , and the unit tangent vector is obtained by choosing an orientation on T . An orientation is arbitrarily chosen at the first iteration while for next iterations orientations are those that minimize the angle between two consecutive tangent vectors. If the followed path is diffeomorphic to a line, the tracking algorithm has to be applied a second time with the opposite orientation.

When the path-tracking process stops because stopping criterion is satisfied, some driving parameters are added or removed. As a consequence, the next part of the path will not be tracked in the same space. To reinitialize the tracking algorithm, one of two orientations on the line T has to be chosen again. Last used tangent vector is translated in the new space in order to compare it with unit vectors of T .

A loop can be detected when a point of the curve is close enough to the initial point $(0, d_{s_k}^+)$. A loop detection test is then performed when hyperplane \mathcal{P}_0 lies between previous and current points. In such a case, we search a point on \mathcal{P}_0 by using the same mechanism as described above to precise a solution on \mathcal{P}_1 and compare it with initial point in the space of original driving parameters.

6.2.2 Differentiation of the CP

When applying the above path-tracking method to track the homotopy path of a function h_b , most of the computation time is spent in the evaluation of the underlying CP. Most evaluations intervene in computation of jacobian matrices by finite difference method that needs about d evaluations, and such matrices are computed at each prediction step and at each iteration of the Newton-Raphson method in a correction step.

It makes the differentiation one of the operations to improve in priority. Here we exploit the acyclic computation scheme of a construction plan to improve it.

Table 1: Running times for an Intel(R) Core(TM) i5 CPU 750 @ 2.67GHz.

Dof/Dor	Octahedron 12	Disulfide 18	Dodecagon 21	Icosahedron 30
HOM4PS2:				
nb sol	4	18	2	-
times	26.7s	6129s	13h	-
first path of H				
nb sol	4	8	2	28
times	0.09s	1.8s	0.15s	9.6s
Presented method				
times	0.035s	0.4s	0.06s	2.8s

Suppose driving parameters $k = (k_0, \dots, k_{d-1})$ appear in steps $I^{i_0}, \dots, I^{i_{d-1}}$ with $i_{d-1} \geq \dots \geq i_0$. The vector $\frac{\partial h_b}{\partial k_{d-1}}(t, k)$ of the jacobian matrix is obtained by evaluating h_b in $(t, k_0, \dots, k_{d-1} - \epsilon)$ and in $(t, k_0, \dots, k_{d-1} + \epsilon)$ for a sufficiently small ϵ . Remark that figures resulting of these evaluations differs only from the step i_{d-1} since $I^{i_{d-1}}$ is the first step that involves k_{d-1} .

As a result, a manner of optimizing the cost of the differentiation in (t, k) of h_b is to evaluate it entirely a first time in (t, k) , to save the obtained figure and hence to compute partial derivatives $\frac{\partial h_b}{\partial k_i}$ in order of decreasing index by evaluating the CP from step I^{k_i} . Our implementation incorporates this optimization.

6.3 Results

We give here numerical results about application of the path tracking mechanism presented in this paper to point-distance problems, one in 2D and three in 3D.

Table 1 presents these results. For each problem, are given the number of equations of the numerical system F (row ‘‘Dof/Dor’’) and running times and number of solutions when applying a classical homotopy method to solve F . It is implemented by the free software HOM4PS-2.0 [18]. Cost in computation time increases dramatically with the degree of freedom of the system due to growing number of path to be followed.

Then row ‘‘first path of H ’’ gives the number of solutions found and times needed to follow the homotopy path when homotopy function used is H defined as a system of equations (so with no triangularization).

Finally, the row ‘‘Presented method’’ gives running times of **Continuation solving of RCP with adaptive CP**. Details about tuning main parameters of this process are presented below before a brief description of its execution for each problem.

6.3.1 Details about parameters

The interpolation function a is defined as follows:

$$a^i(t) = \begin{cases} -t^2 + (a_{st}^i - a_{sk}^i + 1)t + a_{sk}^i & \text{if } i = 0, \\ a_{sk}^i(1-t) + a_{st}^i t & \text{otherwise.} \end{cases}$$

a has a compact positive support and fulfills assumptions of characterization of homotopy paths presented in [9].

For each problem, the prediction step δ of the path tracker has been chosen as the biggest such that the process terminates with success.



Figure 13: Left: the octahedron problem. Right: the disulfide problem. Edges are distance constraints.

Stopping criterion α has been set to value 0.1 that seems to fit well to our algorithm.

6.3.2 The problems

The goal of the octahedron problem is to construct a solid with 6 vertices, 12 edges and 8 triangular faces knowing the lengths of its 12 edges. A quoted sketch of this problem is given in left part of Fig. 13. This problem is related to the parallel robot called Stewart platform and results in a system F of 12 equations involving 12 unknowns. With parameters:

$$\begin{array}{llll} h^0 = 1.11983, & h^1 = 1.17389, & h^2 = 1.18117, & h^3 = 1.06129, \\ h^4 = 1.12482, & h^5 = 1.18592, & h^6 = 1.18592, & h^7 = 1.15857, \\ h^8 = 1.17417, & h^9 = 1.07569, & h^{10} = 1.19071, & h^{11} = 1.16643, \end{array}$$

it admits 4 solutions up to rigid motions. These solutions are found in about 27 seconds by HOM4PS2. When using the sketch of Fig. 13, these 4 solutions lie on the path of the sketch, hence they are all found when following it.

The second problem comes from molecular chemistry and is picked up from [16]. Coordinates of 8 points in the 3D space have to be found knowing 18 pairwise distances. It corresponds to a disulfide molecule (see right-hand part of Fig. 13). A valuation of parameters is exhibited in [16] that leads to 18 solutions all found by a branch-and-prune algorithm in more than 10 minutes. With these parameters, only 8 solutions lie on the path of the sketch but there are found in 0.4s.

Dodecagon and Icosahedron problems are illustrated on Fig. 14. The former gives rise to a numerical function F with 21 constraints, for which 2 solutions were found both by HOM4PS2 and our approach when using parameters:

$$\begin{array}{llll} h^0 = 3, & h^1 = 1.772, & h^2 = 1.7088, & h^3 = 2.05183 \\ h^4 = 1.5, & h^5 = 1.83848, & h^6 = 1.44222, & h^7 = 1.36015 \\ h^8 = 1, & h^9 = 1.38924, & h^{10} = 1, & h^{11} = 0.58309 \\ h^{12} = 4.39318, & h^{13} = 5.10882, & h^{14} = 3.88973, & h^{15} = 3.04138 \\ h^{16} = 3.34215, & h^{17} = 4.39318, & h^{18} = 4.43847, & h^{19} = 6.63702 \\ h^{20} = 4.66154. & & & \end{array}$$

The numerical system of equations associated to the icosahedron problem involves 30 equations and is too large to be solved by a classical homotopy method. With our approach, we find 28 solutions within 4 seconds.

6.3.3 Details on execution

Table 2 presents details about execution of Continuation solving of RCP with adaptive CP for each problem. The rows “Path of H ” recall times of tracking paths of function H and give

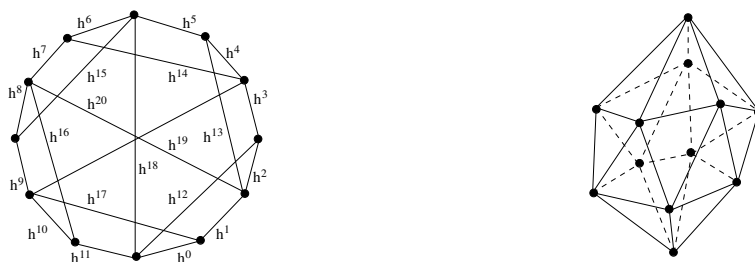


Figure 14: Left: the dodecagon problem. Right: the icosahedron problem. Edges are distance constraints.

Table 2: Details about path-tracking.

Dof/Dor	Octahedron 12	Disulfide 18	Dodecagon 21	Icosahedron 30
Path of H :				
time t in s:	0.09	1.8	0.15	9.6
iterations i :	163	2608	253	3215
t/i in ms:	0.5	0.6	0.6	2,9
Presented method:				
RCP changing:	20	32	2	204
DP: average (max.):	1.4 (2)	2.8 (4)	4 (4)	4.03 (6)
time t in s:	0.035	0.4	0.06	2.8
iterations i :	159	2020	268	6955
t/i in ms:	0.3	0.2	0.2	0.4

the number of iterations of prediction-correction needed. The rows “Presented method” recall running times, give the number of iterations of prediction-correction, the number of changes of the RCP (row “RCP changing”) and finally the average (and maximum) number of driving parameter of the RCP (average is weighted by the number of prediction-correction performed with), in row “DP: average (max.)”.

In both cases, row “ t/i in ms” gives the average time spent in one prediction-correction iteration. It shows how using parameterized construction plans does speed-up the path tracking process.

7 Conclusion

Well-constrained GCSP often have many solutions. The existing solvers offer either a single solution or all the solutions. These latter are of limited practical interest because either the class of problems they solve are reduced or their complexity is exponential. But even if not all solutions are needed, an effective method must provide to the user several solutions close to the sketch.

In this article we present a method for solving geometric constraints problems by homotopy that is effective and robust. Regarding efficiency, the method does not follow all paths but only one starting from the sketch. We then find the nearest solution of the sketch but also other solutions. Indeed we are in the conditions of having a homotopy path diffeomorphic to a circle to which belong usually more than one solution.

Moreover, this method is robust because critical cases, where the path is close to the boundary of the domain of definition, are dealt by a changing of the construction plan. This original idea is sound as shown in the examples discussed. In these examples solutions are produced very quickly compared to other methods that generate multiple solutions.

References

- [1] E.L. Allgower and K. Georg. Numerical path following. *Handbook of Numerical Analysis*, 5(3):207, 1997.
- [2] Daniel J Bates, Jonathan D Hauenstein, Andrew J Sommese, and Charles W Wampler. Adaptive multiprecision path tracking. *SIAM Journal on Numerical Analysis*, 46(2):722–746, 2008.
- [3] C. Durand and C.M. Hoffmann. A systematic framework for solving geometric constraints analytically. *Journal of Symbolic Computation*, 30(5):493–519, 2000.
- [4] D. Faudot and D. Michelucci. A new robust algorithm to trace curves. *Reliable computing*, 13(4):309–324, 2007.
- [5] S. Foufou and D. Michelucci. The Bernstein basis and its applications in solving geometric constraint systems. *Reliable Computing*, 17(2):192–208, 2012.
- [6] Xiao-Shan Gao, Christoph M. Hoffmann, and Wei-Qiang Yang. Solving spatial basic geometric constraint configurations with locus intersection. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*, SMA '02, pages 95–104, New York, NY, USA, 2002. ACM.
- [7] B. Huber and B. Sturmfels. A polyhedral method for solving sparse polynomial systems. *Mathematics of computation*, 64(212):1541–1556, 1995.
- [8] R. Imbach, P. Mathis, and P. Schreck. Tracking method for reparametrized geometrical constraint systems. In *2011 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 31–38. IEEE, 2011.
- [9] Rémi Imbach, Pascal Schreck, and Pascal Mathis. Leading a continuation method by geometry for solving geometric constraints. *Computer-Aided Design*, 46:138–147, 2014.
- [10] R.B. Kearfott and Z. Xing. An interval step control for continuation methods. *SIAM Journal on Numerical Analysis*, 31(3):892–914, 1994.
- [11] Hervé Lamure and Dominique Michelucci. Solving geometric constraints by homotopy. In *Proceedings of the third ACM symposium on Solid modeling and applications*, SMA '95, pages 263–269, New York, NY, USA, 1995. ACM.
- [12] Tiejun Li and Fengshan Bai. Minimizing multi-homogeneous bézout numbers by a local search method. *Mathematics of computation*, 70(234):767–787, 2001.
- [13] Benjamin Martin, Alexandre Goldsztejn, Laurent Granvilliers, and Christophe Jermann. Certified parallelotope continuation for one-manifolds. *SIAM Journal on Numerical Analysis*, 51(6):3373–3401, 2013.
- [14] J. Milnor. Topology from the differentiable viewpoint. *Univ. Press Virginia*, 1965.

- [15] B. Mourrain and J. P. Pavone. Subdivision methods for solving polynomial equations. *J. Symb. Comput.*, 44(3):292–306, March 2009.
- [16] Josep M Porta, Lluís Ros, Federico Thomas, Francesc Corcho, Josep Cantó, and Juan Jesús Pérez. Complete maps of molecular-loop conformational spaces. *Journal of computational chemistry*, 28(13):2170–2189, 2007.
- [17] Andrew John Sommese and Charles Weldon Wampler. *The Numerical solution of systems of polynomials arising in engineering and science*, volume 99. World Scientific, 2005.
- [18] T. Y. Liet T. L. Lee and C. H. Tsai. Hom4ps-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method. *COMPUTING*, 83:109–133, 2008.



**RESEARCH CENTRE
NANCY – GRAND EST**

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399