On solving contact problems with Coulomb friction: formulations and numerical comparisons

Vincent Acary, Maurice Brémond, Olivier Huber

On solving contact problems with Coulomb friction: formulations and numerical comparisons

Vincent Acary, Maurice Brémond, Olivier Huber

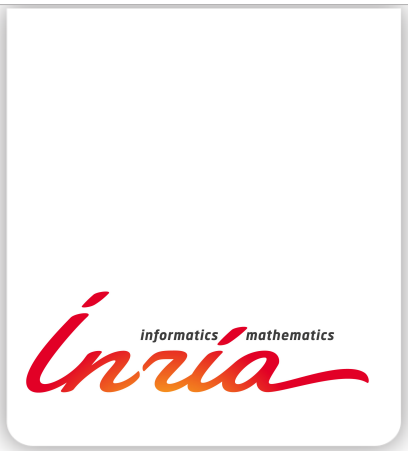
RESEARCH REPORT

N° 9118

November 8, 2017

Project-Team Bipop

ISSN 0249-6399 ISRN INRIA/RR--9118--FR+ENG



On solving contact problems with Coulomb friction: formulations and numerical comparisons

Vincent Acary*†, Maurice Brémond‡†, Olivier Huber§

Project-Team Bipop

Research Report n° 9118 — November 8, 2017 — 224 pages

* vincent.acary@inria.fr
† INRIA, 655, Avenue de l'Europe, 38334 Saint Ismier Cedex, France
‡ maurice.bremond@inria.fr
§ ohuber2@wisc.edu, Wisconsin Institute for Discovery, University of Wisconsin-Madison, 330 N. Orchard St., 53715, USA

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Abstract: In this report, we review several formulations of the discrete frictional contact problem that arises in space and time discretized mechanical systems with unilateral contact and three-dimensional Coulomb's friction. Most of these formulations are well–known concepts in the optimization community, or more generally, in the mathematical programming community. To cite a few, the discrete frictional contact problem can be formulated as variational inequalities, generalized or semi–smooth equations, second–order cone complementarity problems, or as optimization problems such as quadratic programming problems over second-order cones. Thanks to these multiple formulations, various numerical methods emerge naturally for solving the problem. We review the main numerical techniques that are well-known in the literature and we also propose new applications of methods such as the fixed point and extra-gradient methods with self-adaptive step rules for variational inequalities or the proximal point algorithm for generalized equations. All these numerical techniques are compared over a large set of test examples using performance profiles. One of the main conclusion is that there is no universal solver. Nevertheless, we are able to give some hints to choose a solver with respect to the main characteristics of the set of tests

Key-words: Multibody systems, nonsmooth Mechanics, unilateral constraints, Coulomb friction, impact, numerical methods

Sur la résolution du problème de frottement tridimensionnel. Formulations and comparaisons des méthodes numériques.

Résumé : Dans ce rapport, plusieurs formulations du problème discret de contact frottant qui apparaît dans les systèmes mécaniques avec du contact unilatéral et du frottement de Coulomb, sont présentées. La plupart de ces formulations sont des objets bien connus dans la communauté de l'optimisation, et plus généralement, de la programmation mathématique. Pour en citer quelques uns, le probléme de contact frottant peut être formulé comme une inégalité variationnelle, comme une équation non-régulière ou semi–lisse, comme un problème de complémentarité sur des cônes, ou encore comme des problèmes d'optimisation par exemple des problèmes quadratiques sur des cônes du second ordre. Grâce à ces multiples formulations, de nombreuses méthodes numériques de résolutions émergent naturellement. On détaille dans ce rapport les principales techniques numériques bien connues dans la litérature et nous proposons aussi des nouvelles méthodes comme les méthodes de point fixe et d'extra-gradient pour les inégalités variationnelles avec une règle d'adaptation automatique du pas, ainsi que l'application de l'algorithme du point optimal pour les équations généralisées. Toutes ces techniques sont comparées sur un grand ensemble de problème–tests en utilisant des profils de performance. Une des conlusions est qu'il n'existe pas de méthode universelle. Néanmoins, on peut donner des conseils pour choisir une méthode particulière la mieux adaptée aux caractéristiques d'un problème donné.

Mots-clés : Systèmes multi–corps, Mécanique non régulière, contraintes unilatérales, frottement de Coulomb, impact, Schémas numériques de résolution

Contents

1 Introduction

More than thirty years after the pioneering work of [Panagiotopoulos, 1975], [Nečas et al., 1980], [Haslinger, 1983, 1984, Haslinger and Panagiotopoulos, 1984], [Del Piero and Maceri, 1983, 1985], [Katona, 1983], [Chaudhary and Bathe, 1986], [Jean and Moreau, 1987], [Mitsopoulou and Doudoumis, 1988] on numerically solving mechanical problems with contact and friction, there are still active research on this subject in the computational mechanics and applied mathematics communities. This can be explained by the fact that problems from mechanical systems with unilateral contact and Coulomb friction are difficult to numerically solve and the mathematical results of convergence of the numerical algorithms are rare and most of these require rather strong assumptions. In this report, we want to give some insights of the advantages and weaknesses of standard solvers found in the literature by comparing them on large sets of examples coming from the simulation of a wide range of mechanical systems. Some new numerical schemes are also introduced, mainly based on general solvers for variational inequalities and the proximal point algorithms.

1.1 Problem statement

In this section, we formulate an abstract, algebraic finite–dimensional frictional contact problem. We cast this problem as a complementarity problem over cones, and discuss the properties of the latter. We end by presenting some instances with contact and friction phenomenon that fits our problem description.

Abstract problem We want to discuss possible numerical solution procedures for the following three–dimensional finite–dimensional frictional contact problem and some of its variants. Let $n_c \in \mathbb{N}$ be the number of contact points and $n \in \mathbb{N}$ the number of degrees of freedom of a discrete mechanical system.

The problem data are: a positive definite matrix $M \in \mathbb{R}^{n\times n}$, a vector $f \in \mathbb{R}^n$, a matrix $H \in \mathbb{R}^{n\times m}$ with $m = 3n_c$, a vector $w \in \mathbb{R}^m$ and a vector of coefficients of friction $\mu \in \mathbb{R}^{n_c}$. The unknowns are two vectors $v \in \mathbb{R}^n$, a velocity–like vector and $r \in \mathbb{R}^m$, a contact reaction or impulse, solution to

$$\begin{cases} Mv = Hr + f \\ K^\star \ni \hat{u} \perp r \in K \end{cases} \quad \text{with} \quad \begin{array}{l} u := H^\top v + w \\ \hat{u} := u + g(u), \end{array} \tag{1}$$

where the set K is the cartesian product of Coulomb's friction cone at each contact, that is

$$K = \prod_{\alpha=1\dots n_c} K^\alpha = \prod_{\alpha=1\dots n_c} \{r^\alpha, \|r_{\mathrm{T}}^\alpha\| \leqslant \mu^\alpha |r_{\mathrm{N}}^\alpha|\} \tag{2}$$

and $K^\star$ is dual cone of K. The function $g\colon \mathbb{R}^m \to \mathbb{R}^m$ is a nonsmooth function defined as

$$g(u) = [[\mu^\alpha \|u_{\mathrm{T}}^\alpha\|, 0, 0]^\top, \alpha = 1 \dots n_c]^\top. \tag{3}$$

Note that the variable u and $\hat{u}$ do not appear as unknowns since they can be directly obtained from v.

A Second Order Cone Complementarity Problem (SOCCP). From the mathematical programming point of view, the problem appears to be a Second Order Cone Complementarity Problem (SOCCP) [Facchinei and Pang, 2003] which can be generically defined as

$$\begin{cases} y = f(x) \\ K^\star \ni y \perp x \in K, \end{cases} \tag{4}$$

where K is a second order cone. If the nonlinear part of the problem (1) is neglected ($g(u) = 0$), the problem is an associated friction problem with dilatation, and by the way, is a gentle Second Order Cone Linear Complementarity Problem (SOCLCP) with a positive definite matrix $W = H^\top M^{-1} H$ (possibly semi–definite). The assumption of an associated frictional law, i.e, a friction law where the local sliding velocity is normal to the friction cone differs dramatically from the standard Coulomb friction since it generates a non–vanishing normal velocity when the system slides. In other terms, the sliding motion implies the separation of the bodies. When the non-associated character of the friction is taken into account through $g(u)$, the problem is non monotone and nonsmooth, and therefore is very hard to solve efficiently. For a given numerical algorithm, it is not so difficult to design mechanical examples to run the algorithm into troubles [Cadoux, 2009].

Proof of convergence of the numerical algorithms are rare and most of these required strong assumptions among the following ones: a) small values of the friction coefficients, b) full rank assumptions and the symmetry of the Delassus matrix W or c) the assumption that the problem is two-dimensional. Among these results, we can cite the Czech school where the coefficient of friction is assumed to be bounded and small. This assumption allows us to use fixed point methods on the convex sub–problems of Tresca friction (friction threshold that does depend on the normal reaction and then transform the cone into a semi-cylinder). We can also mention the results from [Pang and Trinkle, 1996, Stewart and Trinkle, 1996, Anitescu and Potra, 1997] where the friction cone is polyhedral (in 2D or by a faceting process). In that case, if $w = 0$ or $w \in \mathrm{im}(H^\top)$, Lemke's algorithm is able to solve the problem. The question of existence of solutions has also been treated in [Klarbring and Pang, 1998, Acary et al., 2011] recalled in Section 2.3 under similar assumptions but with different techniques. The question of uniqueness remains a difficult problem in the general case.

Range of applicability. We clearly choose to simplify a lot the general problems of formulating the contact problems with friction by avoiding including too much side effects that are themselves interesting but render the study too difficult to carry out in a single report. We choose finite dimensional systems where the time dependency does not appear explicitly. Nevertheless, we believe that there is a strong interest to study this problem since it appears to be relatively generic in numerous simulations of systems with contact and friction. This problem is indeed at the heart of the simulation of mechanical systems with 3D Coulomb's friction and unilateral constraints in the following cases:

- It might be the result of the time–discretization by event–capturing time–stepping methods or event–detecting (event–driven) techniques of dynamical systems with friction; the variables are homogeneous to pairs velocities/impulses or accelerations/forces.
- It might also be the result of space–discretization (by FEM for instance) of the elastic quasi-static problems of frictional contact mechanics; in that case, the variables are homogenous to displacements/forces of displacement rates/forces.
- If the system is a dynamical mechanical system composed of flexible solids, the problem is again obtained by a space and time discretization.
- If the material follows a nonlinear mechanical bulk behavior, we can use this model after a standard Newton linearization procedure.

For a description of the derivation of such problems in various practical situations we refer to [Laursen, 2003, Wriggers, 2006, Acary and Brogliato, 2008, Acary and Cadoux, 2013].

1.2 Objectives and outline of the report

In this report, after stating the problem with more details in Section 2, we recall the existence result of [Acary et al., 2011] for the problem (1) in Section 2.3. In this framework, we briefly present in Section 3 a few alternative formulations of the problem that enable the design of numerical solution procedures : a) finite–dimensional Variational Inequalities(VI) and Quasi-Variational Inequalities(QVI), b) nonsmooth equations and c) optimization based formulations.

Right after these formulations, we list some of the most standard algorithms dedicated to one of the previous formulations :

1. the fixed point and projection numerical methods for solving VI are reviewed with a focus on self-adaptive step rules (Section 4),
2. the nonsmooth (semi-smooth) Newton methods are described based on the various nonsmooth equations formulations (Section 5),
3. Section 6 is devoted to the presentation of splitting and proximal point techniques,
4. and finally, in Section 7, the Panagiotopoulos alternating optimization technique, the successive approximation technique and the SOCLCP approach are outlined.

Since it is difficult to be exhaustive on the approaches developed in the literature to solve frictional contact problems, we decided to leave out the scope of the report the following approaches:

- the approaches that alter the fundamental assumptions of the 3D Coulomb friction model by faceting the cone as in the pioneering work of [Klarbring, 1986] and followed by [Al-Fahed et al., 1991, Pang and Trinkle, 1996, Stewart and Trinkle, 1996, Anitescu and Potra, 1997, Haslinger et al., 2004], or by convexifying the Coulomb law (associated friction law with normal dilatancy) [Heyn et al., 2013, Tasora and Anitescu, 2013, 2011, Anitescu and Tasora, 2010, Tasora and Anitescu, 2009, Krabbenhoft et al., 2012] or finally by regularizing the friction law [Kikuchi and Oden, 1988].
- the recent developments of methods for the frictionless case [Morales et al., 2008, Miyamura et al., 2010, Temizer et al., 2014] will not be discussed.
- the approaches that are based on domain decomposition and parallel computing [Breitkopf and Jean, 1999, Renouf et al., 2004, Koziara and Bićanić, 2011, Wohlmuth and Krause, 2003, Dostál et al., 2010, Heyn, 2013]. We choose in this report to focus on single domain computation and to skip the discussion about distributed computing mainly for the sake of length of the report.

Finally, some possible interesting approaches have not been reported. We are thinking mainly to the interior point methods approach [Christensen and Pang, 1998, Miyamura et al., 2010, Kleinert et al., 2014]. Some basic implementations of such methods do not give satisfactory results. One of the reasons is the fact that we were not able to get robustness and efficiency on a large class of problems. As it is reported in [Kleinert et al., 2014, Krabbenhoft et al., 2012], it seems that it is needed to alter the friction Coulomb's law by adding regularization or dilatency in the model. In the same spirit, we skip also the comparison for the possibly very promising methods developed in [Heyn et al., 2013, Heyn, 2013] that are based on Krylov subspace and spectral methods. It could be very interesting to bench also these methods on the actual Coulomb friction model, that is to say, in the nonmonotone case. Finally, our preliminary results on the use of direct general SOCP or SOCLCP solvers off the shelf were not convincing. Indeed, the structure of contact problems (product of a large number of small second order cones) has to be taken into account to get efficiency and unfortunately, these solvers are difficult to adapt to this structure.

Other comparisons have already been published in the literature. One of the first comparison study has been done in [Raous et al., 1988] and in [Chabrand et al., 1998]. In this work, several formulations are detailed in the bidimensional case (variational inequality, linear complementarity problem (LCP) and augmented Lagrangian formulation) and comparisons of fixed point methods with projection, splitting methods and Lemke's method for solving LCP. Other comparisons have been done on 2D systems in [Mijar and Arora, 2000b,a, 2004a,b]. In [Christensen et al., 1998], a very interesting comparison in the three-dimensional case has been carried out which shows the superiority of the semi-smooth Newton methods over the interior point methods. Comparisons on simple multi-body systems composed of kinematic chains can be found in [Mylapilli and Jain, 2017].

As a difference with the previous publications, the comparison are performed on a large set of examples using performance profiles in this report. Let us summarize the main conclusion from Section 8: on one hand, the algorithms based on Newton methods for nonsmooth equations solve quickly the problem when they succeed, but suffer from robustness issues mainly if the matrix H has not full rank. On the other hand, the iterative methods dedicated to solving variational inequalities are quite robust but with an extremely slow rate of convergence. To sum up, as far as we know there is no option that combines time efficiency and robustness. The set of problems used here are from the FCLIB collection[1]. In this work, this collection is solved with the software Siconos and its component Siconos/Numerics[2][Acary et al., 2015].

1.3 Notation

The following notation is used throughout the report: the 2-norm for a function g is denoted by $\|g\|$ and for a vector $x \in \mathbb{R}^n$ by $\|x\|$. The index $\alpha \in \mathbb{N}$ is used to identify the variable pertaining to a single contact. A multivalued mapping $T\colon \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is an operator whose images are sets. The second order cone, also known as Lorentz or ice–cream cone, is defined as $K_\mu \coloneqq \{(x,t) \in \mathbb{R} \times \mathbb{R}_+ \mid \|x\| \leqslant \mu t\}$, $\mu \geqslant 0$. By polarity, the dual convex cone to a convex cone K defined by

$$K^\star = \{x \in \mathbb{R}^n \mid y^\top x \geqslant 0, \quad \text{for all } y \in K\}. \tag{5}$$

The normal cone $N_K\colon \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ to a closed convex set X is the set

$$N_K(x) = \{d \in \mathbb{R}^n \mid d^\top (y - x) \leqslant 0\}. \tag{6}$$

The notation $0 \leqslant x \perp y \geqslant 0$ denotes that $x \geqslant 0$, $y \geqslant 0$ and $x^\top y = 0$. A complementarity problem associated with a function $F\colon \mathbb{R}^n \to \mathbb{R}^n$ is to find $x \in \mathbb{R}^n$ such that $0 \leqslant F(x) \perp x \geqslant 0$. The generalized complementarity problem is given by $K^\star \ni F(x) \perp x \in K$, where K is a closed convex cone. Finite-dimentional Variational Inequality (VI) problems subsumes complementarity problems, system of equations. Solving a $\mathrm{VI}(X, F)$ is to find $x \in X$ such that

$$F(x)^\top (y - x) \geqslant 0 \qquad \text{for all } y \in X. \tag{7}$$

It is easy to see this problem is equivalent to solving a *generalized equation*

$$0 \in F(X) + N_X(x). \tag{8}$$

The Euclidean projector on a set X is denoted by P_X.

[1] `https://frictionalcontactlibrary.github.io/index.html`, which aims at providing many problems to compare algorithms on a fair basis

[2] `http://siconos.gforge.inria.fr`

2 Description of the 3D frictional contact problems

2.1 Signorini's condition and Coulomb's friction.

Let us consider the contact between two bodies $A \subset \mathbb{R}^3$ and $B \subset \mathbb{R}^3$ with sufficiently smooth boundaries, as depicted on Figure 1.

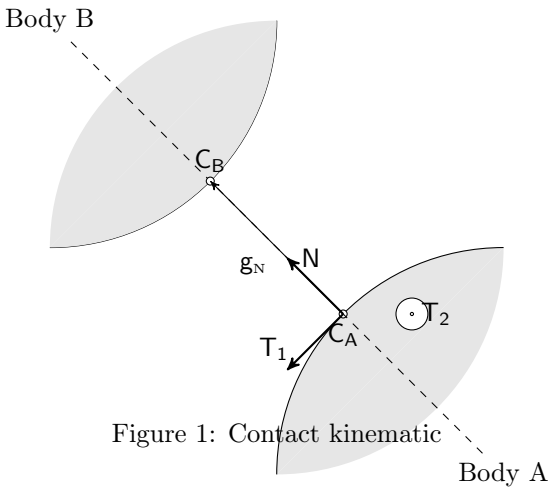


Figure 1: Contact kinematic

From the body A "perspective", the point $C_A \in \partial A$ is called a *master point to contact*. The choice of this master point C_A to write the contact condition is crucial in practice and amounts to consistently discretizing the contact surface. The vector N defines an outward unit normal vector to A at the point C_A. With $\mathsf{T_1}, \mathsf{T_2}$ two vectors in the plane orthogonal to N, we can build an orthornormal frame $(C_A, \mathsf{N}, \mathsf{T_1}, \mathsf{T_2})$ called the *local frame at contact*. The slave contact point $C_B \in \partial B$ is defined as the projection of the point C_A on ∂B in the direction given by N. Note that we assume that such a point exists. The gap function is defined as the signed distance between C_A and C_B

$$g_{\mathrm{N}} = (C_B - C_A)^\top \mathsf{N}. \tag{9}$$

Consider two strictly convex bodies, which are non penetrating, *i.e.* $A \cap B = \emptyset$, the master and slave contact points can be chosen as the proximal points of each bodies and the normal vector N can be written as

$$\mathsf{N} = \frac{C_B - C_A}{\|C_B - C_A\|}. \tag{10}$$

The contact force exerted by A on B is denoted by $r \in \mathbb{R}^3$ and is decomposed in the local frame as

$$r \coloneqq r_{\mathrm{N}}\mathsf{N} + r_{\mathrm{T}_1}\mathsf{T_1} + r_{\mathrm{T}_2}\mathsf{T_2}, \quad \text{with } r_{\mathrm{N}} \in \mathbb{R} \text{ and } r_{\mathrm{T}} \coloneqq [r_{\mathrm{T}_1}, r_{\mathrm{T}_2}]^\top \in \mathbb{R}^2. \tag{11}$$

The *Signorini condition* states that

$$0 \leqslant g_{\mathrm{N}} \perp r_{\mathrm{N}} \geqslant 0, \tag{12}$$

and models the unilateral contact. The condition (12), written at the *position level*, can also be defined at the *velocity level*. To this end, the relative velocity $u \in \mathbb{R}^3$ of the point C_B with respect to C_A is also decomposed in the local frame as

$$u \coloneqq u_{\mathrm{N}}\mathsf{N} + u_{\mathrm{T}_1}\mathsf{T_1} + u_{\mathrm{T}_2}\mathsf{T_2} \quad \text{with } u_{\mathrm{N}} \in \mathbb{R} \text{ and } u_{\mathrm{T}} = [u_{\mathrm{T}_1}, u_{\mathrm{T}_2}]^\top \in \mathbb{R}^2. \tag{13}$$

At the velocity level, the Signorini condition is written

$$\begin{cases} 0 \leqslant u_{\mathrm{N}} \perp r_{\mathrm{N}} \geqslant 0 & \text{if } g_{\mathrm{N}} \leqslant 0 \\ r_{\mathrm{N}} = 0 & \text{otherwise.} \end{cases} \tag{14}$$

The Moreau's viability Lemma [Moreau, 1988] ensures that (14) implies (12) if $g_{\mathrm{N}} \geqslant 0$ holds in the initial configuration.

In some mechanical problems, especially the rigid multi-body systems dynamics, an impact law has to be introduced to complete the dynamics. The most simple law is the Newton impact law that relates the post impact velocity u_{N} to the pre-impact velocity u_{N}^- through a coefficient of restitution $e \geqslant 0$ as

$$u_{\mathrm{N}} = -eu_{\mathrm{N}}^-. \tag{15}$$

Following the work of J.J. Moreau [Moreau, 1988], the impact law is embedded in the Signorini condition at the velocity level as

$$\begin{cases} 0 \leqslant u_{\mathrm{N}} + eu_{\mathrm{N}}^- \perp r_{\mathrm{N}} \geqslant 0 & \text{if } g_{\mathrm{N}} \leqslant 0 \\ r_{\mathrm{N}} = 0 & \text{otherwise.} \end{cases} \tag{16}$$

where r_{N} plays the role of an impulse. The pre-impact velocity is a known value, and then, can be treated as a constant term in w of equation (1). For the sake of simplicity, we will consider in the sequel that $-eu_{\mathrm{N}}^-$ is included in the vector w.

Coulomb's friction models the frictional behavior of the contact force law in the tangent plane spanned by (T_1, T_2). Let us define the Coulomb friction cone K which is the isotropic second order cone (Lorentz or ice–cream cone)

$$K = \{r \in \mathbb{R}^3 \mid \|r_{\mathrm{T}}\| \leqslant \mu r_{\mathrm{N}}\}, \tag{17}$$

where μ is the coefficient of friction. The Coulomb friction states for the sticking case that

$$u_{\mathrm{T}} = 0, \quad r \in K, \tag{18}$$

and for the sliding case that

$$u_{\mathrm{T}} \neq 0, \quad r \in \partial K, \quad \text{and} \quad \exists\, \alpha > 0 \text{ such that } r_{\mathrm{T}} = -\alpha u_{\mathrm{T}}. \tag{19}$$

With the Coulomb friction model, there are two relations between u_{T} and r_{T}. The distinction is based on the value of the relative velocity u_{T} between the two bodies. If $u_{\mathrm{T}} = 0$ (sticking case), we have $\|r_{\mathrm{T}}\| \leqslant \mu r_{\mathrm{N}}$. On the other hand, we get the sliding case.

Disjunctive formulation of the Signorini-Coulomb model If we consider the velocity-level Signorini condition (14) together with the Coulomb friction (18)–(19) which is naturally expressed in terms of velocity, we obtain a disjunctive formulation of the frictional contact behavior as

$$\begin{cases} r = 0 & \text{if } g_{\mathrm{N}} > 0 \quad \text{(no contact)} \\ r = 0, u_{\mathrm{N}} \geqslant 0 & \text{if } g_{\mathrm{N}} \leqslant 0 \quad \text{(take–off)} \\ r \in K, u = 0 & \text{if } g_{\mathrm{N}} \leqslant 0 \quad \text{(sticking)} \\ r \in \partial K, u_{\mathrm{N}} = 0, \exists\, \alpha > 0, u_{\mathrm{T}} = -\alpha r_{\mathrm{T}} & \text{if } g_{\mathrm{N}} \leqslant 0 \quad \text{(sliding)} \end{cases} \tag{20}$$

In the computational practice, the disjunctive formulation is not suitable for solving the Coulomb problem as it suggests the use of enumerative solvers, with an exponential complexity. In the sequel, alternative formulations of the Signorini-Coulomb model suitable for numerical applications are delineated. The core idea is to translate the cases in (20) into complementarity relations.

Inclusion into normal cones The Signorini condition (12) and (14), in their complementarity forms can be equivalently written as an inclusion into a normal cone to $\mathbb{R}_+$

$$-g_{\mathrm{N}} \in N_{\mathbb{R}_+}(r_{\mathrm{N}}) \qquad \text{and} \qquad -u_{\mathrm{N}} \in N_{\mathbb{R}_+}(r_{\mathrm{N}}), \tag{21}$$

if $g_{\mathrm{N}} \leqslant 0$ and $r_{\mathrm{N}} = 0$ otherwise. An inclusion form of the Coulomb friction for the tangential part can be also proposed: let $D(c)$ be the disk of radius c:

$$D(c) \coloneqq \{x \in \mathbb{R}^2 \mid \|x\| \leqslant c\}. \tag{22}$$

For the Coulomb friction, we get

$$-u_{\mathrm{T}} \in N_{D(\mu r_{\mathrm{N}})}(r_{\mathrm{T}}). \tag{23}$$

Since $D(\mu r_{\mathrm{N}})$ is not a cone, the inclusion (23) is not a complementarity problem, but a variational inequality. The formulation (23) is often related to Moreau's maximum dissipation principle of the frictional behavior:

$$r_{\mathrm{T}} \in \underset{\|z\| \leqslant \mu r_{\mathrm{N}}}{\arg\max}\ z^{\top} u_{\mathrm{T}}. \tag{24}$$

This means that the couple $(r_{\mathrm{T}}, u_{\mathrm{T}})$ maximizes the energy lost through dissipation.

SOCCP formulation of the Signorini-Coulomb model In [Acary and Brogliato, 2008, Acary et al., 2011], another formulation is proposed inspired by the so-called bipotential [De Saxcé, 1992, De Saxcé and Feng, 1991, Saxcé and Feng, 1998]. The goal is to form a complementarity problem out of (21) and (23). To this end, we introduce the modified relative velocity $\hat{u} \in \mathbb{R}^3$ defined by

$$\hat{u} = u + [\mu \|u_{\mathrm{T}}\|, 0, 0]^{\top}. \tag{25}$$

The entire contact model (20) can be put into a Second-Order Cone Complementarity Problem (SOCCP) as

$$K^{\star} \ni \hat{u} \perp r \in K \tag{26}$$

if $g_{\mathrm{N}} \leqslant 0$ and $r = 0$ otherwise.

2.2 Frictional contact discrete problems

We assume that a finite set of n_c contact points and their associated local frames have been defined. In general, this task is not straightforward and amounts to correctly discretizing the contact surfaces. For more details, we refer to [Wriggers, 2006, Laursen, 2003]. For each contact $\alpha \in \{1, \ldots, n_c\}$, the local velocity is denoted by $u^{\alpha} \in \mathbb{R}^3$, the normal velocity by $u^{\alpha}_{\mathrm{N}} \in \mathbb{R}$ and the tangential velocity by $u^{\alpha}_{\mathrm{T}} \in \mathbb{R}^2$ with $u^{\alpha} = [u^{\alpha}_{\mathrm{N}}, (u^{\alpha}_{\mathrm{T}})^{\top}]^{\top}$. The vectors $u, u_{\mathrm{N}}, u_{\mathrm{T}}$ respectively collect all the local velocities $u = [(u^{\alpha})^{\top}, \alpha = 1 \ldots n_c]^{\top}$, all the normal velocities $u_{\mathrm{N}} = [u^{\alpha}_{\mathrm{N}}, \alpha = 1 \ldots n_c]^{\top}$, and all the tangential velocities $u_{\mathrm{T}} = [(u^{\alpha}_{\mathrm{T}})^{\top}, \alpha = 1 \ldots n_c]^{\top}$. For a contact α, the modified local velocity, denoted by $\hat{u}^{\alpha}$, is defined by

$$\hat{u}^{\alpha} = u^{\alpha} + g^{\alpha}(u) \qquad \text{where} \qquad g^{\alpha}(u^{\alpha}) = [\mu^{\alpha} \|u^{\alpha}_{\mathrm{T}}\|, 0, 0]^{\top}. \tag{27}$$

The vector $\hat{u}$ and the function g collect all the modified local velocity at each contact $\hat{u} = [\hat{u}^{\alpha}, \alpha = 1 \ldots n_c]^{\top}$ and the function $g(u) = [[\mu^{\alpha} \|u^{\alpha}_{\mathrm{T}}\|, 0, 0]^{\top}, \alpha = 1 \ldots n_c]^{\top}$.

For each contact α, the reaction vector $r^{\alpha} \in \mathbb{R}^3$ is also decomposed in its normal part $r^{\alpha}_{\mathrm{N}} \in \mathbb{R}$ and the tangential part $r^{\alpha}_{\mathrm{T}} \in \mathbb{R}^2$ as $r^{\alpha} = [r^{\alpha}_{\mathrm{N}}, (r^{\alpha}_{\mathrm{T}})^{\top}]^{\top}$. The Coulomb friction cone for a contact α is defined by $K^{\alpha} = \{r^{\alpha} \in \mathbb{R}^3 \mid \|r^{\alpha}_{\mathrm{T}}\| \leqslant \mu^{\alpha} |r^{\alpha}_{\mathrm{N}}|\}$ and the set $K^{\alpha,\star}$ is its dual. The set K is the cartesian product of Coulomb's friction cone at each contact, that is

$$K = \prod_{\alpha=1,\ldots,n_c} K^{\alpha} \quad \text{and} \quad K^{\star} \text{ is its dual.} \tag{28}$$

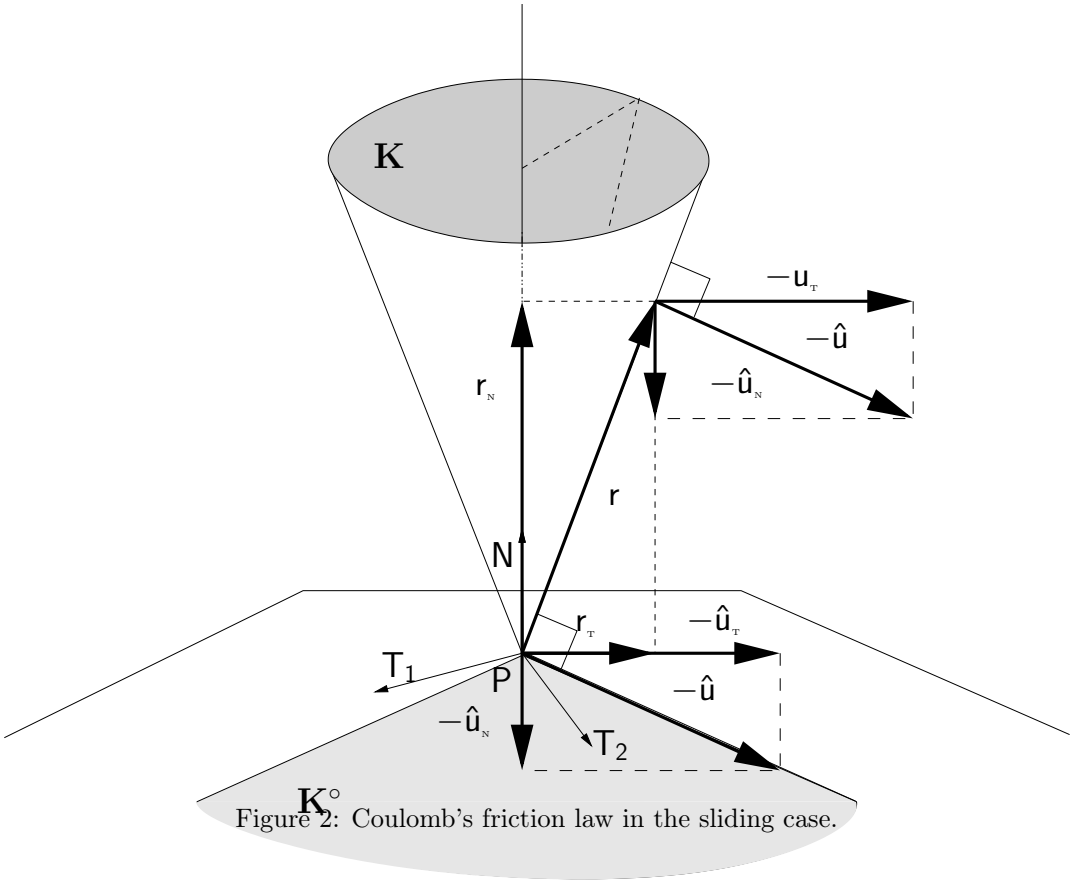


Figure 2: Coulomb's friction law in the sliding case.

In this report, we investigate the case where the problem is given in its *reduced* form. We consider that the discretized and linearized dynamics is of the form

$$Mv = Hr + f, \tag{29}$$

with M a positive-definite matrix. The local velocities at the point of contact are given by

$$u = H^\top v + w. \tag{30}$$

More information on the term w is given later in this section. The (global) velocities v can be substituted in (30) by using a Schur-complement technique. This yields

$$u = H^\top M^{-1} Hr + H^\top M^{-1} f + w. \tag{31}$$

Let us define W, often called the *Delassus matrix*, as

$$W \coloneqq H^\top M^{-1} H \tag{32}$$

and the vector q as

$$q \coloneqq H^\top M^{-1} f + w. \tag{33}$$

We are now ready to define the mathematical problem we want to solve.

Problem FC (Discrete frictional contact problem)**.** *Given*

- *a positive semi–definite matrix* $W \in \mathbb{R}^{m \times m}$ *called the Delassus matrix,*

- *a vector* $q \in \mathbb{R}^m$,
- *a vector* $\mu \in \mathbb{R}^{n_c}$ *of coefficients of friction,*

find a vector $r \in \mathbb{R}^m$, *such that*

$$\begin{cases} K^\star \ni \hat{u} \perp r \in K \\ u = Wr + q \\ \hat{u} = u + g(u) \end{cases} \tag{34}$$

with $g(u) = [[\mu^\alpha \|u_T^\alpha\|, 0, 0]^\top, \alpha = 1 \dots n_c]^\top$.
An instance of the problem is denoted by $\mathrm{FC}(W, q, \mu)$ □

Remark 1. *We do not assume that the Delassus matrix* W *is symmetric in the general case. In most of the applications, the Delassus matrix is symmetric since it represents either a mass matrix or a stiffness matrix. Nevertheless, in the rigid body applications, or more generally when large rotations are taken into account, the Delassus matrix is not symmetric. Indeed, in an implicit time-discretization, the Jacobian matrix of the gyroscopic forces brings a skew symmetric matrix in the Delassus matrix.*

2.3 Existence of solutions

The question of the existence of solution for the Problem FC has been studied in [Klarbring and Pang, 1998] and [Acary et al., 2011] with different analysis techniques under the assumption that the Delassus matrix is symmetric. The key assumption for existence of solutions in both articles is as follows

$$\exists v \in \mathbb{R}^m \ : \ H^\top v + w \in \operatorname{int} K^\star, \tag{35}$$

or equivalently

$$w \in \operatorname{im} H + \operatorname{int} K^\star. \tag{36}$$

Under the previous assumption, the Problem FC have a solution. Therefore, it makes sense to design a procedure to solve the problem. In the sequel, we will compare numerical methods only when this assumption is satisfied.

This assumption is easily verified in numerous applications. For applications in nonsmooth dynamics where the unknown v is a relative contact velocity, the term w vanishes if we have only scleronomic constraints. For $w \in \operatorname{im}(H^\top)$ (and especially $w = 0$), the assumption is trivially satisfied. As it is explained in [Acary and Cadoux, 2013], the term w has several possible sources. If the constraints are formulated at the velocity level, an input term of w is given in dynamics by the impact laws (see equation (16)). In the case of the Newton impact law, it holds that $w \in \operatorname{im}(H^\top)$. For other impact laws, this is not clear. Another input in w is given by constraints that depend explicitly on time. In that case, we can have $w \notin \operatorname{im}(H^\top)$ and non existence of solutions. If the constraints are written at the position level, w can be given by initial terms that come from the velocity discretization. In that cases, the existence is also not ensured.

The assumption is also satisfied whenever $\operatorname{im} H = \mathbb{R}^m$ or in other words if $H^\top$ has full row rank. Unfortunately, in large number of applications $H^\top$ is rank deficient. From the mechanical point of view, the rank deficiency of H and the amount of friction seems to play a fundamental role in the question of the existence (and uniqueness) of solutions. In the numerical comparisons, we will attempt to get a deeper understanding on the role of these assumptions on the convergence of the algorithms. The rank deficiency of H is related to the number of constraints that are imposed to the system with respect to

the number of degrees of freedom in the system. It is closely related to the concept of hyperstaticity in overconstrained mechanical systems. In the most favorable cases, it yields indeterminate Lagrange multipliers but also to unfeasible problems and then to the lost of solutions in the worse cases. The second assumption on the amount of friction is also well–known. The frictionless problem is easy to solve if it is feasible. It is clear that large friction coefficients prevent from sliding and therefore increase the degree of hyperstaticity of the system.

3 Alternative formulations

In this section, various equivalent formulations of Problem FC are given. Our goal is to show that such problems can be recast into several well-known problems in the mathematical programming and optimization community. These formulations will serve as a basis for numerical solution procedures that we develop in later sections.

3.1 Variational Inequalities (VI) formulations

Let us recall the definition a finite-dimensional VI(X, F): find $z \in X$ such that

$$F^\top(z)(y-z) \geqslant 0 \quad \text{for all } y \in X, \tag{37}$$

with X a nonempty subset of $\mathbb{R}^n$ and F a mapping from $\mathbb{R}^n$ into itself. We refer to [Harker and Pang, 1990, Facchinei and Pang, 2003] for the standard theory of finite–dimensional variational inequalities. The easiest way to state equivalent VI formulations of Problem FC is to use the following equivalences:

$$K^\star \ni \hat{u} \perp r \in K \quad \Longleftrightarrow \quad -\hat{u} \in N_K(r) \quad \Longleftrightarrow \quad \hat{u}^\top(s-r) \geqslant 0, \text{ for all } s \in K. \tag{38}$$

For Problem FC, the following equivalent formulation in VI is directly obtained from

$$-\left(Wr+q+g(Wr+q)\right) \in N_K(r). \tag{39}$$

The resulting VI is denoted by VI$(F_{\text{vi}}, X_{\text{vi}})$ with

$$F_{\text{vi}}(r) \coloneqq Wr+q+g(Wr+q) \quad \text{and} \quad X_{\text{vi}} \coloneqq K. \tag{40}$$

Uniqueness properties. In the general case, it is difficult to prove uniqueness of solutions to (40). If the matrix H has full rank and the friction coefficients are "small", a classical argument for the uniqueness of solution of VIs can be satisfied. Note that the full rank hypothesis on H implies that W is positive-definite. Therefore, we have $(x-y)^\top W(x-y) \geqslant C_W\|x-y\|^2$ with $C_W > 0$. Using this relation (40), it yields

$$\begin{array}{rcl} (F_{\text{vi}}(x)-F_{\text{vi}}(y))^\top(x-y) & = & (x-y)^\top W(x-y) \\ & & \quad + \sum_{\alpha=1}^{n_c} \mu^\alpha (x_{\text{N}}^\alpha - y_{\text{N}}^\alpha)[\|[Wx+q]_{\text{T}}^\alpha\| - \|[Wy+q]_{\text{T}}^\alpha\|] \\ & \geqslant & C_W\|x-y\|^2 + \sum_{\alpha=1}^{n_c} \mu^\alpha (x_{\text{N}}^\alpha - y_{\text{N}}^\alpha)[\|[Wx+q]_{\text{T}}^\alpha\| - \|[Wy+q]_{\text{T}}^\alpha\|]. \end{array} \tag{41}$$

Note that for small values of the coefficients of friction the first term in the right-hand side dominates the second one. Hence, the mapping F_{vi} is strictly monotone and this ensures that the VI has at most one solution [Facchinei and Pang, 2003, Theorem 2.3.3]. The fact that H is full rank also implies that the Assumption (35) for the existence of solutions is trivially satisfied. Hence, there exists a unique solution to the VI$(F_{\text{vi}}, X_{\text{vi}})$.

3.2 Quasi-Variational Inequalities (QVI)

Let us recast Problem FC into the QVI framework. A QVI is a generalization of the VI, where the feasible set is allowed to depend on the solution. Let us define this precisely: let X be a multi-valued mapping $\mathbb{R}^n \rightrightarrows \mathbb{R}^n$ and let F be a mapping from $\mathbb{R}^n$ into itself. The quasi-variational inequality problem, denoted by QVI(X, F), is to find a vector $z \in X(z)$ such that

$$F^\top(z)(y - z) \geqslant 0, \, \forall y \in X(z). \tag{42}$$

The QVI formulation of the frictional contact problems is obtained by considering the inclusions (21) and (23). We get

$$u^\top(s - r) \geqslant 0, \text{ for all } s \in C(r_{\mathrm{N}}), \tag{43}$$

where $C(r_{\mathrm{N}})$ is the Cartesian product of the semi–cylinders of radius $\mu^\alpha r_{\mathrm{N}}^\alpha$ defined as

$$C(r_{\mathrm{N}}) \coloneqq \prod_{\alpha=1}^{n_c} \left\{ s \in \mathbb{R}^3 \mid s_{\mathrm{N}} \geqslant 0, \|s_{\mathrm{T}}\| \leqslant \mu^\alpha r_{\mathrm{N}}^\alpha \right\}. \tag{44}$$

Note that the QVI (43) involves only u and not $\hat{u}$: this is the main interest of this formulation. The price to pay is the dependence on r of the set $C(r_{\mathrm{N}})$. Problem FC can be expressed as a QVI by substituting the expression of u, which yields

$$(Wr + q)^\top(s - r) \geqslant 0, \text{ for all } s \in C(r_{\mathrm{N}}). \tag{45}$$

This expression is compactly rewritten as QVI$(F_{\mathrm{qvi}}, X_{\mathrm{qvi}})$, with

$$F_{\mathrm{qvi}}(r) \coloneqq Wr + q \quad \text{and} \quad X_{\mathrm{qvi}}(r) \coloneqq C(r_{\mathrm{N}}). \tag{46}$$

Since W is assumed to be positive semi-definite matrix, F_{qvi} is monotone. Thus we get an affine monotone QVI$(F_{\mathrm{qvi}}, X_{\mathrm{qvi}})$ for Problem FC.

3.3 Nonsmooth Equations

In this section, we expose a classical approach to solving a VI or a QVI, based on a reformulation of the inclusion as a nonsmooth equation. The term nonsmooth equation highlights that the mapping we consider fails to be differentiable. This is the price to pay for this reformulation. We can apply fixed-point and Newton-like algorithms to solve the resulting equation. Given the nonsmooth nature of the problem, applying Newton's method appears challenging, but it can still be done for some reformulations. More precisely for Problem FC, we search for an equation of the type

$$G(r) = 0 \tag{47}$$

where G is generally only locally Lipschitz continuous. The mapping G is such that the zeroes of (47) are the solutions of (34).

Natural and normal maps for the VI formulations A general-purpose reformulation of VI is obtained by using the normal and natural maps, see [Facchinei and Pang, 2003] for details. The natural map $F^{\mathsf{nat}}\colon \mathbb{R}^n \to \mathbb{R}^n$ associated with the VI (37) is defined by

$$F^{\mathsf{nat}}(z) \coloneqq z - P_X(z - F(z)), \tag{48}$$

where P_X is the Euclidean projector on the set X. A well-known result (see [Facchinei and Pang, 2003]) states that the solutions of a VI are related to the zeroes of the natural map :

$$z \text{ solves } \mathrm{VI}(X,F) \quad \Longleftrightarrow \quad F^{\mathsf{nat}}(z) = 0, \tag{49}$$

Using (37), it is easy to see that if z solves $\mathrm{VI}(X,F)$, then it is also a solution to $\mathrm{VI}(X,\rho F)$ for any $\rho > 0$. Therefore, we can define a parametric variant of the natural map by

$$F^{\mathsf{nat}}_{\rho}(z) = z - P_X(z - \rho F(z)). \tag{50}$$

The relations given in (49) continue to hold for the parametric mapping. Using those equivalences, the frictional contact problem can be restated as zeroes of nonsmooth functions. With the natural map, Problem FC under the VI form (40) can be reformulated as

$$F^{\mathsf{nat}}_{\mathrm{vi}}(r) \coloneqq \left[\ r - P_K\left(r - \rho(Wr + q + g(Wr+q))\right)\ \right] = 0. \tag{51}$$

Following the same lines, the normal map may also be used to derive algorithms. The normal map $F^{\mathsf{nor}}\colon \mathbb{R}^n \to \mathbb{R}^n$ is defined by

$$F^{\mathsf{nor}}(x) \coloneqq F(P_X(x)) + x - P_X(x), \tag{52}$$

and its parametric variant

$$F^{\mathsf{nor}}_{\rho}(x) = \rho F(P_X(x)) + x - P_X(x). \tag{53}$$

An equivalent result holds

$$z \text{ solves } \mathrm{VI}(X,F) \quad \Longleftrightarrow \quad z = P_X(x) \text{ for some } x \text{ such that } F^{\mathsf{nor}}(x) = 0. \tag{54}$$

The normal map based formulation of VI is also obtained in the same way.

In the seminal work of [Sibony, 1970], iterative methods for solving monotone VIs are based on the natural map and fixed point iterations. The role of ρ is recognized to be very important for the rate of convergence. To improve the methods, Sibony [1970] proposes to use "skewed" projector based on a non-Euclidean metric. Given a positive definite matrix $R \in R^{n\times n}$, a skewed projector $P_{X,R}$ onto X is defined as follows: $z = P_{X,R}(x)$ is the unique solution of the convex programm

$$\begin{cases} \min\ \dfrac{1}{2}(y-x)^{\top} R(y-x), \\ \ s.t. \quad y \in X. \end{cases} \tag{55}$$

The skew natural map can be also defined and yields the following nonsmooth equation

$$F^{\mathsf{nat}}_{R}(z) = z - P_{X,R}(z - R^{-1}F(z)). \tag{56}$$

The zeros of $F^{\mathsf{nat}}_{R}(z)$ are also solution of the $\mathrm{VI}(X,F)$. Considering the skew natural map, we obtain for Problem FC under the VI form (40),

$$F^{\mathsf{nat}}_{\mathrm{vi},R}(r) \coloneqq \left[\ r - P_{K,R}\left(r - R^{-1}(Wr + q + g(Wr+q))\right)\ \right]. \tag{57}$$

The previous case is retrieved by choosing $R = \rho^{-1} I_{n\times n}$.

Jean–Moreau's and Alart-Curnier's functions Using the alternative inclusions formulations (21)–(23) with a given set of parameters $\rho_{\mathrm{N}}, \rho_{\mathrm{T}}$ such that

$$\begin{cases} -\rho_{\mathrm{N}} u_{\mathrm{N}} \in N_{\mathbb{R}^{n_c}_+}(r_{\mathrm{N}}), & \rho_{\mathrm{N}} > 0, \\ -\rho_{\mathrm{T}} u_{\mathrm{T}} \in N_{D(\mu(r_n)_+)}(r_{\mathrm{T}}), & \rho_{\mathrm{T}} > 0, \end{cases} \tag{58}$$

we can replace P_K into $P_{\mathbb{R}^{n_c}_+}$ and $P_{D(\mu(r_n)_+)}$ where

$$D(\mu(r_n)_+) = \prod_{\alpha=1\ldots n_c} D(\mu^\alpha(r^\alpha_{\mathrm{N}})_+). \tag{59}$$

defines the Cartesian product of the Coulomb disks for each contact. The notation x_+ stands for $x_+ = \max(0, x)$. Using this procedure, Jean and Moreau [1987], Christensen et al. [1998] propose the following nonsmooth equation formulation of the frictional contact condition

$$\begin{cases} r_{\mathrm{N}} - P_{\mathbb{R}^{n_c}_+}(r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) = 0, \\ r_{\mathrm{T}} - P_{D(\mu(r_{\mathrm{N}})_+)}(r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}) = 0. \end{cases} \tag{60}$$

The parameters $\rho_{\mathrm{N}}, \rho_{\mathrm{T}}$ may be also chosen contact by contact. Problem FC is then reformulated as

$$F_{\mathrm{mj}}(r) := \begin{bmatrix} r_{\mathrm{N}} - P_{\mathbb{R}^{n_c}_+}(r_{\mathrm{N}} - \rho_{\mathrm{N}}(Wr+q)_{\mathrm{N}}) \\ r_{\mathrm{T}} - P_{D(\mu(r_{\mathrm{N}})_+)}(r_{\mathrm{T}} - \rho_{\mathrm{T}}(Wr+q)_{\mathrm{T}}) \end{bmatrix} = 0. \tag{61}$$

In the seminal work of Alart & Curnier [Curnier and Alart, 1988, Alart and Curnier, 1991], the augmented Lagrangian approach is invoked (see Remark 3) to obtain a similar formulation motivated by the development of nonsmooth (or generalized) Newton methods (see Section 5.2). To be accurate, the original Alart–Curnier function is given by

$$\begin{cases} r_{\mathrm{N}} - P_{\mathbb{R}^{n_c}_+}(r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) = 0, \\ r_{\mathrm{T}} - P_{D(\mu(r_{\mathrm{N}} - \rho u_{\mathrm{N}})_+)}(r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}) = 0. \end{cases} \tag{62}$$

The difference between (60) and (62) is in the radius of the disk: $D(\mu(r_{\mathrm{N}} - \rho u_{\mathrm{N}})_+)$ rather than $D(\mu(r_{\mathrm{N}})_+)$. Problem FC can be also reformulated as in (61) using (62). This yields

$$F_{\mathrm{ac}}(r) := \begin{bmatrix} r_{\mathrm{N}} - P_{\mathbb{R}^{n_c}_+}(r_{\mathrm{N}} - \rho_{\mathrm{N}}(Wr+q)_{\mathrm{N}}) \\ r_{\mathrm{T}} - P_{D(\mu(r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}})_+)}(r_{\mathrm{T}} - \rho_{\mathrm{N}}(Wr+q)_{\mathrm{T}}) \end{bmatrix} = 0. \tag{63}$$

Remark 2. *From the QVI formulation* (43)*, the following nonsmooth equation can also be written*

$$r = P_{C(r_N)}(r - \rho u) \tag{64}$$

which corresponds to (60)*.*

Remark 3. *In the literature of computational mechanics [Curnier and Alart, 1988, Simo and Laursen, 1992, Alart and Curnier, 1991], very similar expressions are obtained using the concept of augmented Lagrangian functions. This concept introduced in the general framework of Optimization by [Hestenes, 1969] and developed and popularized by [Rockafellar, 1974, 1993] is a strong theoretical tool for analyzing existence and regularity of solutions of constrained optimization problems. Its numerical interest is still a subject of intense debate in the mathematical programming community. In the nonconvex nonsmooth context of frictional contact problems, its invocation is not so clear, but it has enabled the design of robust numerical techniques. Nevertheless, it is worth to note that some of these methods appear as variants of the methods developed to solve variational inequalities in other contexts. The method developed by [Simo and Laursen, 1992] is a dedicated version of fixed point with projection for VI (see Section 1) and the method of [Alart and Curnier, 1991] is a tailored version of semi–smooth Newton methods (see Section 5). Nevertheless, the concept of augmented Lagrangian has never been used in the optimization literature for this purpose.*

Xuewen-Soh-Wanji functions Following the earlier work of [Park and Kwak, 1994] and [Leung et al., 1998], the following function is proposed in [Xuewen et al., 2000]:

$$F_{\text{xsw}}(r) \coloneqq \left[\begin{array}{c} \min(u_{\text{N}}, r_n) \\ \min(\|u_{\text{T}}\|, \mu r_{\text{N}} - \|r_{\text{T}}\|) = 0 \\ |u_{\text{T}_1} r_{\text{T}_2} - u_{\text{T}_2} r_{\text{T}_1} + \max(0, u_{\text{T}_1} r_{\text{T}_1}) = 0 \end{array} \right] = 0. \tag{65}$$

In [Xuewen et al., 2000], the system is solved by a generalized Newton method with a line-search procedure.

Hüeber–Stadler–Wolhmuth functions In [Stadler, 2004, Hüeber et al., 2008], and subsequently in [Koziara and Bićanić, 2008], another function is used to reformulate the problem FC:

$$F_{\text{hsw}}(r) \coloneqq \left[\begin{array}{c} r_{\text{N}} - P_{\mathbb{R}^{n_c}_+}(r_{\text{N}} - \rho_{\text{N}}(Wr + q)_{\text{N}}) \\ \max(\mu(r_{\text{N}} - \rho_{\text{N}} u_{\text{N}}), \|r_t - \rho_t u_{\text{T}}\|) r_{\text{T}} - \mu \max(0, r_n - \rho_{\text{N}} u_{\text{N}})(r_t - \rho_{\text{T}} u_{\text{T}}) \end{array} \right] = 0. \tag{66}$$

In [Hüeber et al., 2008], this function is used considering the constraints at the position level rather than in [Koziara and Bićanić, 2008] the formulation is at the velocity level.

General SOCC-functions More generally, a large family of reformulations of the SOCCP (26) in terms of equations can be obtained by using a so-called Second Order Cone Complementarity (SOCC) function. Let us consider the following SOCCP over a symmetric cone $K^\star = K$. A SOCC-function ϕ is defined by

$$K \ni x \perp y \in K \quad \Longleftrightarrow \quad \phi(x, y) = 0. \tag{67}$$

The frictional contact problem can be written as a SOCCP over symmetric cones by applying the following transformations

$$x = T_x \hat{u} = \begin{bmatrix} \hat{u}_{\text{N}} \\ \mu \hat{u}_{\text{T}} \end{bmatrix} \text{ and } y = T_y r = \begin{bmatrix} \mu r_{\text{N}} \\ r_{\text{T}} \end{bmatrix}. \tag{68}$$

Clearly, the nonsmooth equations of the previous sections provide several examples of SOCC-functions and the natural map offers the simplest one. In [Fukushima et al., 2001], the standard complementarity functions for Nonlinear Complementarity Problems (NCP) such as the celebrated Fischer-Burmeister function are extended to the SOCCP by means of Jordan algebra. Smoothing functions are also given with theirs Jacobians and they studied their properties in view of the application of Newton's method. For the second order cone, the Jordan algebra can be defined with the following non-associative Jordan product

$$x \cdot y = \left[\begin{array}{c} x^\top y \\ y_{\text{N}} x_{\text{T}} + x_{\text{N}} y_{\text{T}} \end{array} \right] \tag{69}$$

and the usual componentwise addition $x + y$. The vector x^2 denotes $x \cdot x$ and there exists a unique vector $x^{1/2} \in K$, the square root of $x \in K$, defined as

$$(x^{1/2})^2 = x^{1/2} \cdot x^{1/2} = x. \tag{70}$$

A direct calculation for the SOC in $\mathbb{R}^3$ yields

$$x^{1/2} = \left[\begin{array}{c} s \\ \dfrac{x_{\text{T}}}{2s} \end{array} \right], \quad \text{where } s = \sqrt{(x_{\text{N}} + \sqrt{x_{\text{N}}^2 - \|x_{\text{T}}\|^2})/2}. \tag{71}$$

We adopt the convention that $0^{1/2} = 0$. The vector $|x| \in K$ denotes $(x^2)^{1/2}$. Thanks to this algebra and its associated operator, the projection onto K can be written as

$$P_K(x) = \frac{x + |x|}{2}. \tag{72}$$

This formula provides a new expression for the natural map and its associated nonsmooth equations. This is exactly what is done in [Hayashi et al., 2005] where the natural map (48) is used together with an expression of the projection operator based on the Jordan algebra calculus. The resulting SOCCP is then solved with a semi–smooth Newton method, and a smoothing parameter can be added.

Most of the calculus in Jordan algebra are based on the spectral decomposition, a basic concept in Jordan algebra, (see [Fukushima et al., 2001] for more details). For $x = (x_{\mathrm{N}}, x_{\mathrm{T}}) \in \mathbb{R} \times \mathbb{R}^2$, the spectral decomposition is defined by

$$x = \lambda_1 u_1 + \lambda_2 u_2, \tag{73}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}$ and $u_1, u_2 \in \mathbb{R}^3$ are the spectral values and the spectral vectors of x given by

$$\lambda_i = x_{\mathrm{N}} + (-1)^i \|x_{\mathrm{T}}\|, \quad u_i = \begin{cases} \frac{1}{2}\begin{bmatrix} 1 \\ (-1)^i \dfrac{x_{\mathrm{T}}}{\|x_{\mathrm{T}}\|} \end{bmatrix}, \text{ if } x_{\mathrm{T}} \neq 0 \\ \frac{1}{2}\begin{bmatrix} 1 \\ (-1)^i w \end{bmatrix}, \text{ if } x_{\mathrm{T}} = 0 \end{cases} \qquad i = 1, 2 \tag{74}$$

with $w \in \mathbb{R}^2$ any unit vector. Note that the decomposition is unique whenever $x_{\mathrm{T}} \neq 0$. The spectral decomposition enjoys very nice properties that simplifies the computation of basic functions such that

$$\begin{aligned} &x^{1/2} = \sqrt{\lambda_1} u_1 + \sqrt{\lambda_2} u_2, \text{ for any } x \in K, \\ &P_K(x) = \max(0, \lambda_1) u_1 + \max(0, \lambda_2) u_2. \end{aligned} \tag{75}$$

More interestingly, general SOCC-functions can also be extended and smoothed version of this function can be also developed (see [Fukushima et al., 2001]). Let us start with the Fischer-Burmeister function

$$\phi_{\mathsf{FB}}(x, y) = x + y - (x^2 + y^2)^{1/2}. \tag{76}$$

It can be shown that the zeroes of ϕ_{FB} are solutions of the SOCCP (67) using the Jordan algebra associated with K. Using the spectral decomposition, the Fischer-Burmeister function can be easily computed as

$$\phi_{\mathsf{FB}}(x, y) = x + y - (\sqrt{\bar{\lambda}_1}\bar{u}_1 + \sqrt{\bar{\lambda}_2}\bar{u}_2) \tag{77}$$

where $\bar{\lambda}_1, \bar{\lambda}_2 \in \mathbb{R}$ and $\bar{u}_1, \bar{u}_2 \in \mathbb{R}^3$ are the spectral values and the spectral vectors of $x^2 + y^2$ that is

$$\begin{aligned} \bar{\lambda}_i &= \|x\|^2 + \|y\|^2 + 2(-1)^i \|x_{\mathrm{N}} x_{\mathrm{T}} + y_{\mathrm{N}} y_{\mathrm{T}}\| \\ \bar{u}_i &= \begin{cases} \frac{1}{2}\begin{bmatrix} 1 \\ (-1)^i \dfrac{x_{\mathrm{N}} x_{\mathrm{T}} + y_{\mathrm{N}} y_{\mathrm{T}}}{\|x_{\mathrm{N}} x_{\mathrm{T}} + y_{\mathrm{N}} y_{\mathrm{T}}\|} \end{bmatrix}, \text{ if } x_{\mathrm{N}} x_{\mathrm{T}} + y_{\mathrm{N}} y_{\mathrm{T}} \neq 0 \\ \frac{1}{2}\begin{bmatrix} 1 \\ (-1)^i w \end{bmatrix}, \text{ if } x_{\mathrm{N}} x_{\mathrm{T}} + y_{\mathrm{N}} y_{\mathrm{T}} = 0 \end{cases} \end{aligned} \quad , i = 1, 2. \tag{78}$$

Finally, Problem FC is then reformulated as

$$F_{\mathsf{FB}}(u, r) := \begin{bmatrix} u - Wr - q \\ \Phi_{\mathsf{FB}}\left(\begin{bmatrix} \mu r_{\mathrm{N}} \\ r_{\mathrm{T}} \end{bmatrix}, \begin{bmatrix} \frac{1}{\mu}(u_{\mathrm{N}} + \mu \|u_{\mathrm{T}}\|) \\ u_{\mathrm{T}} \end{bmatrix}\right) \end{bmatrix} = 0. \tag{79}$$

where the mapping $\Phi_{\mathsf{FB}} : \mathbb{R}^{3n_c} \times \mathbb{R}^{3n_c} \to \mathbb{R}^{3n_c}$ is defined as

$$\Phi_{\mathsf{FB}}(x, y) = \left[(\phi(x^\alpha, y^\alpha), \alpha = 1 \dots n_c)^\top\right]. \tag{80}$$

3.4 Optimization problems

In this section, several optimization-based formulations are proposed. The quest for an efficient optimization formulation of the frictional problem is a hard task. Since the problem is nonsmooth and nonconvex, the use of an associated optimization problem is interesting from the numerical point of view if we want to improve the robustness and the stability of the numerical methods.

A straightforward optimization problem can be written whose cost function is the scalar product $r^\top \hat{u}$. Indeed, this product is always positive and vanishes at the solution. Let us consider this first optimization formulation

$$\begin{cases} \min\ r^\top \hat{u} = r^\top u + \sum_{\alpha=1}^{n_c} \mu^\alpha r_{\mathrm{N}}^\alpha \|u_{\mathrm{T}}^\alpha\| \\ \quad s.t. \quad \hat{u} \in K^\star, \\ \qquad\ \ r \in K, \end{cases} \tag{81}$$

which amounts to minimizing the DeSaxcé's bipotential function [De Saxcé, 1992] over $K^\star \times K$. A first simplification can be made by noting that

$$\hat{u} \in K^\star \Longleftrightarrow u_{\mathrm{N}} \geqslant 0, \tag{82}$$

which leads to

$$\begin{cases} \min\ r^\top u + \sum_{\alpha=1}^{n_c} \mu^\alpha r_{\mathrm{N}}^\alpha \|u_{\mathrm{T}}^\alpha\| \\ \quad s.t. \quad u_{\mathrm{N}} \geqslant 0 \\ \qquad\ \ r \in K. \end{cases} \tag{83}$$

Starting from Problem FC, a direct substitution of $u = Wr + q$ yields

$$\begin{cases} \min\ r^\top (Wr+q) + \sum_{\alpha=1}^{n_c} \mu^\alpha r_{\mathrm{N}}^\alpha \|(Wr+q)_{\mathrm{T}}^\alpha\| \\ \quad s.t. \quad (Wr+q)\mathsf{N} \geqslant 0, \\ \qquad\ \ r \in K. \end{cases} \tag{84}$$

which is a nonlinear optimization problem with a nonsmooth and nonconvex cost function. From the numerical point of view this problem may be very difficult and we have to ensure that the cost function has to be zero at the solution which is not guaranteed if some local minima are reached in the minimization process.

Other optimization-based formulations have been proposed in the literature. They are not direct optimization formulation but they try to identify an optimization sub-problem which is well-posed and for which efficient numerical methods are available. Three approaches can be listed in three categories: a) the *alternating optimization* problems, b) the *successive approximation* method and c) the *convex SOCP* approach.

The Panagiotopoulos alternating optimization approach The Panagiotopoulos alternating optimization approach aims at solving the frictional contact problem by alternatively solving the Signorini condition for a fixed value of the tangential reaction r_{T}, and solving the Coulomb friction model for a fixed value of the normal reaction r_{N}. Let us split the matrix W and the vector q in the following way:

$$u = Wr + q \quad \Longleftrightarrow \quad \begin{bmatrix} u_{\mathrm{N}} \\ u_{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} W_{\mathrm{NN}} & W_{\mathrm{NT}} \\ W_{\mathrm{TN}} & W_{\mathrm{TT}} \end{bmatrix} \begin{bmatrix} r_{\mathrm{N}} \\ r_{\mathrm{T}} \end{bmatrix} + \begin{bmatrix} q_{\mathrm{N}} \\ q_{\mathrm{T}} \end{bmatrix}. \tag{85}$$

Two sub-problems can therefore be identified: the first one is to find $u_{\rm N}$ and $r_{\rm N}$ such that

$$\begin{cases} u_{\rm N} = W_{\rm NN} r_{\rm N} + \tilde{q}_{\rm N}, \\ 0 \leqslant u_{\rm N} \perp r_{\rm N} \geqslant 0, \end{cases} \tag{86}$$

where $\tilde{q}_{\rm N} = q_{\rm N} + W_{\rm NT} r_{\rm T}$. The second problem is to find $u_{\rm T}$ and $r_{\rm T}$ such that

$$\begin{cases} u_{\rm T} = W_{\rm TT} r_{\rm T} + \tilde{q}_{\rm T}, \\ -u_{\rm T} \in N_{D(\mu \tilde{r}_{\rm N})}(r_{\rm T}), \end{cases} \tag{87}$$

where $\tilde{r}_{\rm N}$ is fixed and $\tilde{q}_{\rm T} = q_{\rm T} + W_{\rm TN} r_{\rm N}$.

If we assume for a while that the Delassus W is a symmetric positive semi–definite matrix, $W_{\rm NN}$ and $W_{\rm TT}$ are also symmetric semi–definite positive matrices. Therefore, two convex optimization problems can be formulated:

$$\begin{cases} \min \ \dfrac{1}{2} r_{\rm N}^\top W_{\rm NN} r_{\rm N} + r_{\rm N}^\top \tilde{q}_{\rm N} \\ \text{s.t.} \quad r_{\rm N} \geqslant 0, \end{cases} \tag{88}$$

and

$$\begin{cases} \min \ \dfrac{1}{2} r_{\rm T}^\top W_{\rm TT} r_{\rm T} + r_{\rm T}^\top \tilde{q}_{\rm T} \\ \text{s.t.} \quad r_{\rm T} \in D(\mu \tilde{r}_{\rm N}). \end{cases} \tag{89}$$

This approach has been proposed by [Panagiotopoulos, 1975] for two–dimensional applications in soil foundation computing. It has also been used in other finite element applications in [Barbosa and Feijóo, 1985, Tzaferopoulos, 1993] and studied from the mathematical point of view in [Haslinger and Panagiotopoulos, 1984, Haslinger et al., 1996].

Remark 4. *If the Delassus matrix is unsymmetric matrix but semi-definite positive, the following quadratic programming problem is equivalent to (86)*

$$\begin{cases} \min \ r_N^\top W_{NN} r_N + r_N^\top \tilde{q}_N \\ s.t. \quad r_N \geqslant 0 \\ \qquad W_{NN} r_N + \tilde{q}_N \geqslant 0. \end{cases} \tag{90}$$

The successive approximation The successive approximation method identifies a single optimization problem by introducing a function that maps the normal reaction to itself (or the friction threshold) such that

$$h(r_{\rm N}) = r_{\rm N}. \tag{91}$$

Using this artifact, we can define a new problem from Problem FC such that

$$\begin{cases} \theta = h(r_{\rm N}) \\ u = Wr + q \\ -u_{\rm N} \in N_{\mathbb{R}^{n_c}_+}(r_{\rm N}) \\ -u_{\rm T} \in N_{D(\mu\theta)}(r_{\rm T}). \end{cases} \tag{92}$$

If we assume for a while that the Delassus W is a symmetric positive semi–definite matrix the last three lines are equivalent to a convex optimization problem over the product of semi–cylinders $C(\mu, \theta)$, that is

$$\begin{cases} \theta = h(r_{\rm N}) \\ \begin{cases} \min \ \dfrac{1}{2} r^\top W r + r^\top q \\ \text{s.t.} \quad r \in C(\mu, \theta) \end{cases} \end{cases} \tag{93}$$

The method of successive approximation has been extensively used for proving existence and uniqueness of solutions to the discrete frictional contact problems. We refer to [Haslinger et al., 1996] which summarizes the seminal work of the Czech school [Nečas et al., 1980, Haslinger, 1983, 1984]. We will see in the sequel that this approach also provides us with very efficient numerical solvers in Section 7.2.

The convex SOCP The convex SOCP approach is in the same vein as the previous one, with the difference that a SOCQP sub-problem is identified. To this aim, we augment the problem by introduction an auxiliary variable s, the image of $g(u)$ introduced in (27). We then obtain

$$\begin{cases} s = g(u) \\ \hat{u} = Wr + q + s \\ K^\star \ni \hat{u} \perp r \in K. \end{cases} \tag{94}$$

Since W is a positive semi-definite matrix, a new convex optimization sub-problem can be defined

$$\begin{cases} s = g(u) \\ \begin{cases} \min & \frac{1}{2} r^\top W r + r^\top (q + s) \\ \text{s.t.} & r \in K. \end{cases} \end{cases} \tag{95}$$

This formulation introduced in [Cadoux, 2009] and developed in [Acary and Cadoux, 2013, Acary et al., 2011] has been used to give an existence criteria to the discrete frictional contact problems. Furthermore, this existence criteria can be numerically checked by solving a linear program of second-order cone (SOCLP).

4 Numerical methods for VIs

4.1 Fixed point and projection methods for VI

Starting from the VI formulations (37) or more precisely an associated nonsmooth equation through the natural map,

$$F_R^{\mathsf{nat}}(z) = z - P_{X,R}(z - R^{-1}F(z)). \tag{96}$$

The basic idea of the algorithm is to perform fixed point iterations on the mapping

$$z \mapsto P_{X,R}(z - R^{-1}F(z)), \tag{97}$$

yielding to Algorithm 1 with the specific choice of $R = \rho_k^{-1} I$. The choice of the updating rule of ρ_k is detailed in Section 4.2.

For the formulation (40), the following iterations are performed

$$r_{k+1} \leftarrow P_{K,R}(r_k - R^{-1}(Wr_k + q + g(Wr_k + q))). \tag{98}$$

In the sequel when a parameter ρ is specified, it is assumed that $R = \rho^{-1} I$.

The convergence of such methods are generally shown for strongly monotone VI. In our case, this assumption is not satisfied, but we will see in the sequel that such methods can converge in practice.

Remark 5. *Algorithm 1 with the iteration rule* (98) *and a fixed value of* ρ_k *has been originally proposed in [De Saxcé and Feng, 1991, 1998]. The algorithm is called Uzawa's algorithm by reference*

Algorithm 1 Fixed point iterations for the VI (37)

Require: F, X Data of VI (37)
Require: z_0 initial values
Require: tol > 0 a tolerance value and $\text{iter}_{\max} > 0$ the max number of iterations
Require: ρ_0 initial value for ρ
Ensure: z solution of VI (37)
  $k \leftarrow 0$
  while error $>$ tol and $k < \text{iter}_{\max}$ **do**
    Update the value of ρ_k
    $z_{k+1} \leftarrow P_X(z_k - \rho_k F(z_k))$
    Evaluate error.
    $k \leftarrow k + 1$
  end while
  $z \leftarrow z_k$

to the algorithm due to Uzawa in computing the optimal values of convex program by primal-dual techniques[Glowinski et al., 1976, Fortin and Glowinski, 1983]. Note that the algorithm in [Simo and Laursen, 1992] is similar to the fixed point algorithm with projection though based on augmented Lagrangian concept (see Remark 3).

Extragradient methods The extragradient method [Korpelevich, 1976] is also a well-known method for VI which improves the previous projection method. It can be described as

$$\begin{array}{rcl} \bar{z}_k & \leftarrow & P_X(z_k - \rho F(z_k)) \\ z_{k+1} & \leftarrow & P_X(z_k - \rho\, F(\bar{z}_k)) \end{array} \tag{99}$$

and formally defined in Algorithm 2. The convergence of this method is guaranteed under the following

Algorithm 2 Extragradient method for the VI (37)

Require: F, X Data of VI (37)
Require: z_0 initial values
Require: tol > 0 a tolerance value and $\text{iter}_{\max} > 0$ the max number of iterations
Ensure: z solution of VI (37)
  $k \leftarrow 0$
  while error $>$ tol and $k < \text{iter}_{\max}$ **do**
    Update the value of ρ_k
    $\bar{z}_k \leftarrow P_X(z_k - \rho_k F(z_k))$
    $z_{k+1} \leftarrow P_X(z_k - \rho_k\, F(\bar{z}_k))$
    Evaluate error.
    $k \leftarrow k + 1$
  end while
  $z \leftarrow z_k$

assumptions: there exists a solution and the function F is Lipschitz–continuous and pseudo–monotone.

Name	Algo.	Additional informations
FP-DS	1	iteration rule (98) & fixed ρ
FP-VI-UPK	1 & 3	iteration rule (98) and updating rule (101)
FP-VI-UPTS	1 & 3	iteration rule (98) and updating rule (102)
EG-VI-UPK	2 & 3	iteration rule (99) and updating rule (101)
EG-VI-UPTS	2 & 3	iteration rule (99) and updating rule (102)

Table 2: Namenclature for the algorithms based on VI formulations.

4.2 Self-adaptive step-size rules

A key ingredient in this efficiency and the convergence of the numerical methods for VI presented above is the choice of the sequence $\{\rho_k\}$. A sensible work has been done in the literature mainly motivated by some convergence proofs under specific assumption. Besides the relaxation of the assumption for the convergence, we are interesting in improving the numerical efficiency and robustness. We present in this section, the most popular approach for choosing the sequence $\{\rho_k\}$.

In [Khobotov, 1987], a method is proposed to improve the extragradient method of Korpelevich [1976] by adapting ρ_k in the following way. The goal is to find ρ_k that satisfies

$$0 < \rho_k \leqslant \min\left\{\bar{\rho}, L\frac{\|z_k - \bar{z}_k\|}{\|F(z_k) - F(\bar{z}_k)\|}\right\} \text{ with } L \in (0,1) \tag{100}$$

where $\bar{\rho}$ is the maximum value of ρ_k which is chosen in the light of the specific problem. The objective is to find a coefficient that is bounded by the local Lipschitz constant. The standard way to do that is to use an Armijo–type procedure by successively trying some values of $\rho_k = \bar{\rho}\nu^m$ with $m \in \mathbb{N}$ and $\nu \in (0,1)$, with a typical value of $2/3$. In the original article of [Khobotov, 1987], there is no procedure to size $\bar{\rho}$ or to update it. In [He and Liao, 2002] and in the context of prediction–correction, the authors propose to use the rule $\rho_k = \rho_{k-1}\nu^m$ and if the criteria (100) is largely satisfied for ρ_k, the value is increased. In [Han and Lo, 2002], a similar procedure is used for the extragradient method by adding an increasing step of ρ_k, which is done after the correction as in [He and Liao, 2002]. The criteria (100) is verified by computing the ratio

$$r_k \leftarrow \frac{\rho_k\|F(z_k) - F(\bar{z}_k)\|}{\|z_k - \bar{z}_k\|}. \tag{101}$$

In [Solodov and Tseng, 1996], similar Armijo–like technique is used, and the ratio r_k is computed as follows:

$$r_k \leftarrow \frac{\rho_k(z_k - \bar{z}_k)^\top(F(z_k) - F(\bar{z}_k))}{\|z_k - \bar{z}_k\|^2}. \tag{102}$$

The approach is summarized in Algorithm 3. The parameter L typically chosen around 0.9 is a safety coefficient in the evaluation of ρ_k. The parameter $\mathsf{L}_{\mathsf{min}}$ that triggers an increase of ρ_k is chosen around 0.3.

In [Han and Lo, 2002], the update of the Armijo rule $\rho_k \leftarrow \nu\,\rho_k$ can also be replaced by $\rho_k \leftarrow \nu\,\rho_k \min\{1, 1/r_k\}$ but it appears that this trick does improve the self–adaptive procedure. Other more evolved step–lengths strategies can be found in [Wang et al., 2010] that have been tried in this study.

4.3 Nomenclature

A nomenclature for the algorithms based on the VI formulation is given in Table 1.

Algorithm 3 Updating rule for ρ_k

Require: F, X
Require: Search and safety parameters. $\mathsf{L} \in (0,1), 0 < \mathsf{L}_{\mathsf{min}} < \mathsf{L}, \nu \in (0,1)$
Require: Initial values $\mathsf{z_k} \in \mathsf{X}, \rho_{\mathsf{k}-1} > 0$
  $\rho_{\mathsf{k}} \leftarrow \rho_{\mathsf{k}-1}$
  $\bar{\mathsf{z}}_{\mathsf{k}} \leftarrow \mathsf{P}_{\mathsf{X}}(\mathsf{z_k} - \rho_{\mathsf{k}}\mathsf{F}(\mathsf{z_k}))$
  Evaluate r_k with (101) (or (102))
  while $\mathsf{r_k} > \mathsf{L}$ **do**
    $\rho_{\mathsf{k}} \leftarrow \nu\, \rho_{\mathsf{k}}$
    $\bar{\mathsf{z}}_{\mathsf{k}} \leftarrow \mathsf{P}_{\mathsf{X}}(\mathsf{z_k} - \rho_{\mathsf{k}}\mathsf{F}(\mathsf{z_k}))$
    Evaluate r_k with (101) (or (102))
  end while
  Perform the correction step of extragradient or prediction–correction method.
  if $\mathsf{r_k} < \mathsf{L}_{\mathsf{min}}$ **then**
    $\rho_{\mathsf{k}} = \frac{1}{\nu}\rho_{\mathsf{k}}$
  end if

5 Newton based methods

5.1 Principle of the nonsmooth Newton methods

In Section 3.3, several formulations of the frictional contact problem by means of nonsmooth equations have been presented. These nonsmooth equations call for the use of nonsmooth Newton's methods. Remember that the standard Newton method consists in solving

$$G(z) = 0 \tag{103}$$

by performing the following Newton iteration

$$z_{k+1} = z_k - J^{-1}(z_k)G(z_k). \tag{104}$$

If the mapping G is smooth, the matrix J is the Jacobian matrix of G with respect to z, that is $J(z) = \nabla_z G(z)$. Whenever G is nonsmooth but locally Lipschitz continuous, the Jacobian matrix J is replaced by an element $\Phi(z)$ of the generalized Jacobian at z: $\Phi(z) \in \partial G(z)$. Let us recall the definition of the generalized Jacobian. By Rademacher's Theorem, if G is locally Lipschitz continuous, then G is almost everywhere differentiable and let us define the set D_G by

$$D_G \coloneqq \{z \mid G \text{ is differentiable at } z\}. \tag{105}$$

The generalized Jacobian of G at z can be defined by

$$\partial G(z) = \mathrm{conv}\partial_B G(z), \tag{106}$$

with

$$\partial_B G(z) = \{\lim_{\bar{z}\to z, \bar{z}\in D_G} \nabla G(\bar{z})\}. \tag{107}$$

If $\Phi(z)$ is nonsingular, then an iteration of the nonsmooth Newton method is given by

$$z_{k+1} = z_k - \Phi^{-1}(z_k)(G(z_k)). \tag{108}$$

Algorithm 4 Nonsmooth Newton method for (103)

Require: G data of Problem (103)
Require: z_0 initial values
Require: tol > 0 a tolerance value and $\mathsf{iter}_{\max} > 0$ the max number of iterations
Ensure: z solution of Problem (103)
  $\mathsf{k} \leftarrow 0$
  while error $>$ tol and $\mathsf{k} < \mathsf{iter}_{\max}$ **do**
    compute (select) $\Phi(\mathsf{z}_k) \in \partial \mathsf{G}(\mathsf{z}_k)$
    $\mathsf{z}_{k+1} \leftarrow \mathsf{z}_k - \Phi^{-1}(\mathsf{z}_k)(\mathsf{G}(\mathsf{z}_k))$
    Evaluate error.
    $\mathsf{k} \leftarrow \mathsf{k} + 1$
  end while
  $\mathsf{z} \leftarrow \mathsf{z}_k$

The resulting nonsmooth Newton method is detailed in Algorithm 4.

The convergence of nonsmooth Newton methods is based on the assumption of semi–smoothness of the nonsmooth function in (103). For this reason they are often called *semi–smooth Newton methods* (see [Facchinei and Pang, 2003, Section 7.5] and references therein).

5.2 Application to the discrete frictional contact problem

We use the Alart–Curnier function $F_{\mathrm{ac}}(u,r)$ in (63), Jean–Moreau function $F_{\mathrm{mj}}(u,r)$ in (61), Fischer-Burmeister function $F_{\mathsf{FB}}(u,r)$ in (79), and the natural map $F_{\mathrm{vi}}^{\mathsf{nat}}$ in (51) to define a Newton method for the Problem FC.

Computation of an element of ∂G For any r_0 in the nonsmooth domain of G, we compute $\Phi(r_0) = \lim_{t\to 0} \Phi(r(t))$ with $t \to r(t)$ a parametrization such that $\lim_{t\to 0} r(t) = r_0$ with $r(t)$ in the smooth domain for all t. Similar computations can also be found in [Joli and Feng, 2008] where a Newton method based on the formulation (51) is used contact by contact in a Gauss–Seidel loop.

Lipschitz continuity properties For the mappings $F_{\mathrm{vi}}^{\mathsf{nat}}, F_{\mathrm{ac}}, F_{\mathsf{FB}}, F_{\mathrm{mj}}, F_{\mathrm{xsw}}$, whose expressions is mostly made of the Lipschitz functions $P_X, \min, \max$ and $\|.\|$; the local Lipschitz properties can be shown without difficulties. For the mapping F_{FB}, the proof of Lipschitz continuity of ϕ_{FB} can be found in [Sun and Sun, 2005] and references therein. This ensures the consistency of the definition of the generalized Jacobians.

5.3 Convergence and robustness issues.

The local convergence of the nonsmooth Newton methods is based on the semi-smoothness of the mapping G and the fact that all elements of the generalized Jacobian at the solution point $z^\star$, $\Phi(z^\star) \in \partial G(z^\star)$ are non singular (see [Qi and Sun, 1993] and Chapter 1 of [Qi et al., 2018] for a survey of mathematical results). For our application, the semi-smoothness of the mapping $F_{\mathrm{ac}}, F_{\mathrm{mj}},$ or F_{hsw} is proven in several papers [Christensen and Pang, 1998, Hüeber et al., 2008]. The strong semi–smoothness of ϕ_{FB} can be found in [Sun and Sun, 2005].

On the other hand, the regularity of all elements of the generalized Jacobians is not guaranteed. The first reason is the possible rank deficiency of the matrix W, which is usual in rigid body applications

as discussed in Section 2.3. Even if we consider a full rank matrix W, as in the standard one contact case for instance, the invertibility of all the elements of the generalized Jacobian at the solution point is not straightforward. For the mapping $F_{\text{ac}}, F_{\text{mj}}$, some results are given in [Alart, 1993, 1995, Jourdan et al., 1998]. Some of the results depend on the value of the coefficient of friction and the exact penalty parameters $\rho, \rho_{\text{N}}, \rho_{\text{T}}$ parameters. For the mapping F_{hsw}, some other results can be found in [Hüeber et al., 2008].

In the numerical practice and even if W is full-rank, it may happen that the elements of the generalized Jacobians are not regular or very badly conditioned when we are far from the solution. This fact is reported in [Alart, 1993, 1995, Jourdan et al., 1998, Hüeber et al., 2008, Koziara and Bićanić, 2008]. Some divergence of the Newton algorithm can be encountered. A few work has been done to understand this problem. Among them, we cite [Hüeber et al., 2008] where some modifications of the elements of the generalized Jacobian are performed far form the solution to keep the Newton iteration matrix regular and well conditioned when the function F_{hsw} is chosen. This very interesting work opens new directions of research for the other mappings. In [Koziara and Bićanić, 2008], some other heuristics are developed to try to avoid divergence of the Newton loop. In the two next sections, we present two complementary ways to partly solve this problem by choosing consistently the parameters $\rho, \rho_{\text{N}}, \rho_{\text{T}}$ and by applying some line–searches techniques to globalize the convergence.

5.4 Estimation of $\rho, \rho_{\text{N}}, \rho_{\text{T}}$ parameters

One of the key parameters in the efficiency of the nonsmooth Newton methods is the choice of the parameter ρ in the parameterized natural map (50) and the parameters ρ_{N} and ρ_{T} in the Jean–Moreau and Alart-Curnier functions (61) and (63). The default choice is to set these parameters equal to 1 but the numerical practice shows that the convergence of the nonsmooth solvers is drastically deteriorated, especially if the norm or the conditioning of the matrix W is far from this unit value. There is no theoretical rules to size this parameters, but some heuristics may be found in the literature for a single contact problem that we expose in the sequel.

Inverse of a norm of W A first simple choice is to consider the inverse of a norm of the matrix W. With this heuristics, we set the ρ parameter before the Newton loop as follows:

$$\rho = \frac{1}{\|W\|}, \quad \rho_{\text{N}} = \rho_{\text{T}} = \frac{1}{\|W\|}. \tag{109}$$

This choice is mainly based on a guess of the inverse of the local Lipschitz constant of the operator $Wr + q$. In the case of the natural map, it amounts to neglecting the nonlinear contribution of g. For the norm, whenever the matrix is symmetric definite positive, choosing the 2-norm based on the spectral radius $\|W\|_2 = \rho(W) = \lambda_{\max}(W)$ would yield:

$$\rho = \frac{1}{\lambda_{\max}(W)}, \quad \rho_{\text{N}} = \rho_{\text{T}} = \frac{1}{\lambda_{\max}(W)}. \tag{110}$$

Estimation based on the splitting W_{NN} and W_{TT} A second possible choice for the map (61) and (63) is to use the fact that the problem is split with respect to the normal and the tangent directions. In that case, we compute a value of ρ_{N} that is based on the eigenvalues of W_{NN} and a value of ρ_{T} based on the eigenvalue of W_{TT}. For a single contact, we set

$$\rho_{\text{N}} = \frac{1}{W_{\text{NN}}}, \quad \rho_{\text{T}} = \frac{1}{\lambda_{\max}(W_{\text{TT}})} \tag{111}$$

A third option it also to take into account the conditioning of the matrix W_{TT} by choosing

$$\rho_{\mathrm{N}} = \frac{1}{W_{\mathrm{NN}}}, \quad \rho_{\mathrm{T}} = \frac{\lambda_{\min}(W_{\mathrm{NN}})}{\lambda^2_{\max}(W_{\mathrm{TT}})} \tag{112}$$

Again, these heuristics implicitly assume that the Delassus matrix W is symmetric definite positive.

Adaptive estimation of the parameters In [Koziara and Bićanić, 2008], an adaptive way of updating ρ is proposed that has not been implemented for our experiments.

Default choices By default, we use the rule (111) for the mapping (61) and (63) and the rule (110) for the natural map. When other rules are chosen in the comparison, they are specified.

5.5 Damped Newton and line-search procedures

We use mainly two type of line-search procedures: the Goldstein-Price and the Armijo line-search. Usually, strong mathematical assumptions are needed to guarantee their success, especially on the smoothness of the merit function $\mathcal{M}(x)$. For the Newton method, we use as merit function the half of the norm of G, that is

$$\mathcal{M}(x) = \frac{1}{2}\|G(x)\| \tag{113}$$

with G taken accordingly to the formulation equals to $F^{\mathsf{nat}}_{\mathrm{vi}}, F_{\mathrm{ac}}, F_{\mathsf{FB}}, F_{\mathrm{mj}}, F_{\mathrm{xsw}}$. Clearly, the smoothness assumptions are not satisfied in our case. Even if the assumptions are fulfilled and despite the mathematical proofs, in practice, is is recommended that some additional stopping criteria be added during extrapolation and interpolation phases to avoid infinite loops. In the sequel, we use the recommendations in Chapter 3 of [Bonnans et al., 2003], where the reader can find all the mathematical explanations of why they terminate, under some assumptions on the merit function. The choice of the values for the parameters m_1, m_2 for the Goldstein-Price line-search and the parameter m_1 alone for the Armijo line-search is also discussed and it is advised to choose $m_1 < \frac{1}{2}$ and $m_2 > \frac{1}{2}$.

Termination requires the existence of a function $q \in C^1(\mathbb{R})$ with $q'(0) < 0$ which is the value of the merit function in a given direction d. This function has to be bounded from below. In our case, this function is $q : t \to \frac{1}{2}\|G(r + td)\|$. An additional stopping criterion is implemented as a maximum number of iterations and when the line-search fails, the Newton loop is continued with the last value of the step found by the line–search.

The Goldstein–Price (GP) line search and Armijo line search are described in Algorithm 5 and Algorithm 6.

5.6 Nomenclature

A nomenclature for the algorithms based on the nonsmooth Newton methods is listed in Table 2.

6 Splitting techniques and proximal point algorithm

Splitting techniques are standard techniques to solve VI(F, X) when the function F is affine that is $F(z) = Mz + q$ and the set X can decomposed in a Cartesian product of independent smaller sets $X = \Pi_i X_i$. Usually, a block splitting of the matrix M is performed and a Projected Successive Over Relaxation (PSOR) method is used to solve the VI. Since the cone K is a product of second-order cones in $\mathbb{R}^3$, a natural way to split the problem is to form sub-problems by using single contact as a building

Algorithm 5 Goldstein–Price (GP) line search

Require: x, the starting point of the line-search.
Require: d, the direction of search.
Require: t, an initial stepsize-value.
Require: $t \to q(t)$, for $t \geqslant 0$, with $q \in C^1$ bounded from below and $q'(0) < 0$, a merit function representing $f(x + td)$
Require: m_1, m_2, parameters with $0 < m_1 < m_2 < 1$
Require: a, with $a > 1$, parameter for extrapolation
Ensure: a finite line-search

```
t_L ← 0
t_R ← 0
Δ ← (q(t)−q(0))/t
while m_2 q'(0) > Δ or Δ > m_1 q'(0) do
  if m_1 q'(0) < Δ then
    t_R ← t
  end if
  if Δ < m_2 q'(0) then
    t_L ← t
  end if
  if t_R = 0 then
    t ← at
  else
    t ← (t_L + t_R)/2
  end if
  Δ ← (q(t)−q(0))/t
end while
```

Algorithm 6 Armijo(A) line search

Require: x, the starting point of the line-search.
Require: d, the direction of search.
Require: t, an initial stepsize-value.
Require: $t \to q(t)$, for $t \geqslant 0$, with $q \in C^1$ bounded from below and $q'(0) < 0$, a merit function representing $f(x + td)$
Require: m_1, a parameter with $0 < m_1 < 1$
Require: a, with $a > 1$, parameter for extrapolation
Ensure: a finite line-search

```
while m_1 q'(0) < (q(t)−q(0))/t do
  if t_R = 0 then
    t ← at
  else
    t_R ← t
    t ← t_R/2
  end if
end while
```

Name	Algo.	Additional informations
NSN-NM	4	Natural map formulation (51)
NSN-AC	4	Alart–Curnier formulation (63)
NSN-JM	4	Jean–Moreau formulation (61)
NSN-FB	4	Fischer-Burmeister formulation (79)
NSN-NM-GP	4 & 5	Natural map formulation (51) and the Goldstein–Price (GP) line search
NSN-AC-GP	4 & 5	Alart–Curnier formulation (63) and the Goldstein–Price (GP) line search
NSN-JM-GP	4 & 5	Jean–Moreau formulation (61) and the Goldstein–Price (GP) line search
NSN-FB-GP	4 & 5	Fischer-Burmeister formulation (79) and the Goldstein–Price (GP) line search
NSN-NM-A	4 & 6	Natural map formulation (51) and the Armijo(A) line search
NSN-AC-A	4 & 6	Alart–Curnier formulation (63) and the Armijo(A) line search
NSN-JM-A	4 & 6	Jean–Moreau formulation (61) and the Armijo(A) line search
NSN-FB-A	4 & 6	Fischer-Burmeister formulation (79) and the Armijo(A) line search
NSN-AC-HYBRID	4 & 2	Alart–Curnier formulation (63) with a pre computation of the initial guess with 100 iterations of EG-VI-UPK algorithm the

TABLE 2. Naming convention for the algorithms based on nonsmooth Newton (NSN) method

block. The sub-problems can be solved by any method for the VI that have been presented in the previous sections. In the same way, the proximal point algorithm can also be used which amounts to solving the original VI(F, X) by solving a sequence of (easier) VIs.

6.1 Splitting and relaxation techniques

The particular structure of the cone K as a product of second-order cones in $\mathbb{R}^3$ calls for a splitting of the problem contact by contact. For Problem FC, the relation

$$u = Wr + q \tag{114}$$

is splitted along each contact as follows:

$$u^\alpha = W^{\alpha\alpha} r^\alpha + \sum_{\beta \neq \alpha} W^{\alpha\beta} r^\beta + q^\alpha, \qquad \text{for all } \alpha \in 1 \dots n_c, \tag{115}$$

where the matrices α and β are used to label the variable for each contact. The matrices $W^{\alpha\beta}$ with $\alpha \in 1, \dots, n_c$ and $\beta \in 1, \dots, n_c$ are easily identified from (114). From (115), a projected Gauss–Seidel (PGS) method is obtained by using the following update rule at the k-th iterate:

$$u^\alpha_{k+1} = W^{\alpha\alpha} r^\alpha_{k+1} + \sum_{\beta < \alpha} W^{\alpha\beta} r^\beta_{k+1} + \sum_{\beta > \alpha} W^{\alpha\beta} r^\beta_k + q^\alpha, \text{ for all } \alpha \in 1 \dots n_c. \tag{116}$$

A Projected Successive Over Relaxation (PSOR) scheme is derived by introducing a relaxation parameter $\omega > 0$ such that

$$u^\alpha_{k+1} = \frac{1}{\omega} W^{\alpha\alpha} r^\alpha_{k+1} - \frac{1}{\omega} W^{\alpha\alpha} r^\alpha_k + \sum_{\beta < \alpha} W^{\alpha\beta} r^\beta_{k+1} + \sum_{\beta \geqslant \alpha} W^{\alpha\beta} r^\beta_k + q^\alpha, \text{ for all } \alpha \in 1 \dots n_c. \tag{117}$$

At the k-th iteration, the following problem is solved for each contact α:

$$\begin{cases} u^\alpha_{k+1} = \bar{W}^{\alpha\alpha} r^\alpha_{k+1} + \bar{q}^\alpha_{k+1}, \\ \hat{u}^\alpha_{k+1} = u^\alpha_{k+1} + g(u^\alpha_{k+1}), \\ K^{\alpha,\star} \ni \hat{u}^\alpha_{k+1} \perp r^\alpha_{k+1} \in K^\alpha, \end{cases} \tag{118}$$

where

$$\begin{cases} \bar{W}^{\alpha\alpha} = \frac{1}{\omega} W^{\alpha\alpha} \\ \bar{q}^{\alpha}_{k+1} = -\frac{1}{\omega} W^{\alpha\alpha} r^{\alpha}_{k} + \sum_{\beta<\alpha} W^{\alpha\beta} r^{\beta}_{k+1} + \sum_{\beta\geqslant\alpha} W^{\alpha\beta} r^{\beta}_{k} + q^{\alpha} \end{cases} \text{, for all } \alpha \in 1 \dots n_c. \tag{119}$$

The problem (118) has exactly the same structure as Problem FC, but is of lower size since it is only for one contact. It is solved by a *local solver*, which can be any of the algorithms presented in this report or even an analytical method (enumerating all the possible cases as in [Bonnefon and Daviet, 2011]).

The PSOR algorithm is summarized in Algorithm 7 and the NSGS correspond to the case $\omega = 1$.

Algorithm 7 PSOR algorithm for Problem FC

Require: $\mathsf{W}, \mathsf{q}, \mu$
Require: r_0 initial values
Require: $\mathsf{tol} > 0, \mathsf{tol}_{\mathsf{local}}$ tolerance values and $\mathsf{iter}_{\max} > 0, \mathsf{iter}_{\mathsf{local}\max} > 0$ the max number of local iterations
Require: ω a relaxation parameter
Ensure: r, u solution of Problem FC
  while error $>$ tol and $\mathsf{k} < \mathsf{iter}_{\max}$ **do**
    for $\alpha = 1 \dots \mathsf{n_c}$ **do**
      $\bar{\mathsf{W}}^{\alpha\alpha}_{\mathsf{k}+1} \leftarrow \frac{1}{\omega}\mathsf{W}^{\alpha\alpha}$
      $\bar{\mathsf{q}}^{\alpha}_{\mathsf{k}+1} \leftarrow -\frac{1}{\omega}\mathsf{W}^{\alpha\alpha}\mathsf{r}^{\alpha}_{\mathsf{k}} + \sum_{\beta<\alpha}\mathsf{W}^{\alpha\beta}\mathsf{r}^{\beta}_{\mathsf{k}+1} + \sum_{\beta\geqslant\alpha}\mathsf{W}^{\alpha\beta}\mathsf{r}^{\beta}_{\mathsf{k}} + \mathsf{q}^{\alpha}$
      Solve the single contact problem $\mathrm{FC}(\bar{\mathsf{W}}^{\alpha\alpha}, \bar{\mathsf{q}}^{\alpha}_{\mathsf{k}+1}, \mu)$ at accuracy $\mathsf{tol}_{\mathsf{local}}$ with a maximum of iteration $\mathsf{iter}_{\mathsf{local}\max}$
    end for
    Evaluate error.
    $\mathsf{k} \leftarrow \mathsf{k} + 1$
  end while
  $\mathsf{r} \leftarrow \mathsf{r}_{\mathsf{k}}$
  $\mathsf{u} \leftarrow \mathsf{u}_{\mathsf{k}}$

Applications in frictional contact date back to the work of [Mitsopoulou and Doudoumis, 1988, 1987] for two-dimensional friction. In [Jourdan et al., 1998], this method is developed in the Gauss-Seidel configuration ($\omega = 1$) with a local Newton solver based on the Alart–Curnier formulation. If the local solver performs only one iteration of the VI solver based on projection, we get a standard splitting technique for VI. In Table 3, the methods based on PSOR used in the comparison are summarized.

6.2 Proximal points techniques

The first use of the proximal idea dates back to the early days of convex analysis [Moreau, 1965]. The proximity operator of a proper, lower semi-continuous function f is defined as

$$\mathrm{prox}_f(x) = \min_z f(z) + \tfrac{\alpha}{2}\|z - x\|^2, \quad \alpha > 0 \tag{120}$$

and the point $\mathrm{prox}_f(x)$ is called the *proximal point*. The latter is unique whenever f is convex. Recently, there has been a surge in the use of the proximity operator in optimization. There has been applications to non-differentiable, large-scale optimization, mainly since the proximity operator enjoys nicer property: it is differentiable and it may be easier to compute in some cases. There is a wealth of literature on the use of proximal mapping in optimization [Parikh et al., 2014]. The basic idea is to replace (part of)

the objective function by the proximal operator of it. Starting from an initial x_0, a proximal algorithm produces a sequence $\{x_k\}$ by the relation $x_{k+1} = \text{prox}_f(x_k)$. The sequence is guaranteed to converge whenever f is convex. This basic algorithm can be enhanced by a proper choice of the parameter α: some acceleration techniques ensure the convergence of the sequence $\{x_k\}$ with a different α at each iteration. In the non-convex case, the mapping prox_f is still well behaved whenever f is said to be *prox-regular* and α is small enough.

The proximal mapping can also be defined for set-valued mappings. Then it correspond to a regularization of the (sub-) differential of f. More precisely, it correspond to the *resolvent* of ∇f, defined as $R_\alpha := \alpha(\alpha I + T)^{-1}$. Starting from a initial guess x_0, a sequence is computed as $x_{k+1} = R_\alpha(x_k)$ (assuming the singlevaluedness of R_α). Much less attention has been given to this kind of algorithms, especially few numerical studies have been conducted. From the theoretical point of view, the convergence is shown in [Rockafellar, 1976] when the mapping T is maximal monotone, an extension of the convex case previously mentioned. With the same hypothesis, the mapping R_α is single-valued for all $\alpha > 0$. For concreteness, consider the variational inequality

$$0 \in F(x) + N_X(x) \qquad \text{also written as} \quad 0 \in T(x). \tag{121}$$

Then, the proximal point algorithm applied to this VI consists in solving at each step the VI

$$0 \in F(x) + \alpha I - \alpha x_k + N_X(x) \tag{122}$$

that can be compactly written as

$$0 \in F_{\alpha,x_k}(x) + N_X(x). \tag{123}$$

The parameter α can be changed for each sub-VI.

Other variants of the basic algorithm can be derived, like adding a relaxation parameter ω:

$$x_{k+1} = (1-\omega)x_k + \omega z_{k+1}, \tag{124}$$

where z_{k+1} The algorithm is described in algorithm 8.

Algorithm 8 Proximal point algorithm for the VI (37)

Require: F, X Data of VI (37)
Require: ω relaxation parameter
Require: α_0 the initial value of the proximal point parameter
Require: x_0 initial value
Require: tol > 0, tol_{in} tolerance values and $\text{iter}_{\max} > 0$ the max number of iterations
Ensure: x solution of VI (37)
  $k \leftarrow 0$
  while error $>$ tol and $k < \text{iter}_{\max}$ **do**
    Solve VI(F_{α_k,x_k}, X) for z_{k+1} at accuracy tol_{int}
    $x_{k+1} \leftarrow (1-\omega)x_k + \omega z_{k+1}$
    Evaluate error.
    Compute α_{k+1}
    $k \leftarrow k+1$
  end while
  $x \leftarrow x_{k+1}$

For solving the sub-problems VI(F_{α,x_k}, X), any of the previous algorithms for VI can be used. The main interest of the proximal point algorithm is that the mapping F_{α,x_k} is nicer than F. For instance, if $F(x) = Mx+q$, then the matrix in $F(\alpha, x_k)$ is $M+\alpha I$. It is easy to see that for α large enough, $M+\alpha I$ is positive-definite, with no assumption on M. With nonlinear operator, choosing α large enough ensures that F_{α,x_k} is monotone (with some condition on F). In practice, this implies that a greater number of algorithms are able to solve the VI. It is a good indicator of an easier problem to solve, and we observe that this approach is able to provide some robustness to the VI-based approaches. The introduction of two additional parameters (α and $\mathsf{tol}_{\mathsf{int}}$) is the main drawback of this approach. Indeed, instead of solving just one VI, this approach calls for solving multiple sub-VIs. This additional computational effort can be reduced in two ways: the first one is to drive the proximal parameter α_k as quickly as possible to zero, in order to reduce the number of sub-VI to solve. The other option is to set the tolerance $\mathsf{tol}_{\mathsf{int}}$ to a higher value when α_k is large, so as to reduce the computational effort for the sub-VIs. The choice of $\mathsf{tol}_{\mathsf{int}}$ is discussed in Section 6.3.

6.3 Control of the tolerance of internal solvers $\mathsf{tol}_{\mathsf{int}}$ and $\mathsf{tol}_{\mathsf{local}}$ in the splitting and proximal approaches

In Algorithms 7 and 8 an internal tolerance is used to control the required accuracy of the internal solver. It is generally not useful to solve the internal problem at the accuracy of the global one. For Algorithm 7, the local tolerance $\mathsf{tol}_{\mathsf{local}}$ is set by default to a very low value of 10^{-14}. An adaptive local tolerance strategy has also been tested that sets the local tolerance to a fraction of the current error as, for instance, $\mathsf{tol}_{\mathsf{local}} = \mathsf{error}/10$. For the proximal point algorithm in Algorithm 8, the internal tolerance $\mathsf{tol}_{\mathsf{int}}$ is set to a fraction of the error $\mathsf{tol}_{\mathsf{int}} = \mathsf{error}/10$.

6.4 Control of the proximal point parameter α_k

In Algorithm 8, the proximal point parameter α_k is updated for each sub-VI. We choose to implement two rules for its computation. The first one is inspired by the work in [Hager and Zhang, 2008] that is based on the current error or residual of the algorithm. The parameter is computed thanks to the following rule:

$$\alpha_k = \sigma(\mathsf{error})^{\nu}, \tag{125}$$

where $\sigma > 0, \nu > 0$ are two additional parameters that influence the rate of driving α_k to zero. The other rule is an heuristic rule that starts from a given value of α_0. If the internal solver for the sub-VI succeeds to reach the required accuracy then α_{k+1} is decreased and set to $\alpha_{k+1} = \alpha_k/10$. If the internal solver do not succeed then we increase α_{k+1} as $\alpha_{k+1} = 5\alpha_k$.

6.5 Nomenclature

A nomenclature for the algorithms based on the projection/splitting approach is given in Table 3.

7 Optimization based methods

In this section, the Delassus matrix is assumed to be symmetric in order to be able to state simple convex optimization problems.

Name	Algorithm	Additional informations
NSGS-AC	7 with $\omega = 1$	local solver: NSN-AC with tolerance $\mathsf{tol}_{\mathsf{local}}$
NSGS-JM	7 with $\omega = 1$	local solver: NSN-JM with tolerance $\mathsf{tol}_{\mathsf{local}}$
NSGS-AC-GP	7 with $\omega = 1$	local solver: NSN-AC-GP with tolerance $\mathsf{tol}_{\mathsf{local}}$
NSGS-JM-GP	7 with $\omega = 1$	local solver: NSN-JM=GP with tolerance $\mathsf{tol}_{\mathsf{local}}$
NSGS-FP-DS-One	7 with $\omega = 1$	local solver: one iteration of FP-DS
NSGS-FP-VI-UPK	7 with $\omega = 1$	local solver: FP-VI-UPK with tolerance $\mathsf{tol}_{\mathsf{local}}$
NSGS-EXACT	7 with $\omega = 1$	exact local solver
PSOR-AC	7	local solver: NSN-AC with tolerance $\mathsf{tol}_{\mathsf{local}}$
PPA-NSN-AC	8	internal solver: NSN-AC
PPA-NSGS-AC	8	internal solver: NSGS-AC

Table 3: Naming convention for the algorithms based on splitting and proximal algorithms

7.1 Alternating optimization problem

The Panagiotopoulos approach described in Section 3.4 generates a family of solvers by choosing two specific solvers for the normal contact problem (88) and the tangential contact problem (89), respectively. This method may be viewed as a two-block Gauss-Seidel method (as pointed out by [Tzaferopoulos, 1993]). More precisely, the following choices may be made for the normal and tangent problems.

The normal contact problem

$$\begin{cases} \min \dfrac{1}{2} r_{\mathrm{N}}^{\top} W_{\mathrm{NN}} r_{\mathrm{N}} + r_{\mathrm{N}}^{\top} \tilde{q}_{\mathrm{N}} \\ \text{s.t.} \quad r_{\mathrm{N}} \geqslant 0 \end{cases} \qquad \text{with } \tilde{q}_{\mathrm{N}} = q_{\mathrm{N}} + W_{\mathrm{NT}} r_{\mathrm{T},k}, \tag{126}$$

is a convex quadratic program with simple bound constraints. In the literature, a large number of solvers has been developed to solve such problems. Among others, we might cite the active set strategy solvers [Fletcher, 1987, Nocedal and Wright, 1999] that are mainly dedicated to small–scale systems, the projected gradient [Calamai and More, 1987] and projected conjugate gradient methods [Moré and Toraldo, 1989, Moré and Toraldo, 1991] that are more dedicated to large–scale systems. Note that there exists also a wealth of methods in the literature that improves the methods of [Moré and Toraldo, 1991] for large–scale systems. For the reader interested in those details, we refer to the book of [Dostál, 2016](see especially the Section 8.7 for a review of the different approaches). It is clear that we might also use semi-smooth Newton methods or interior point methods but our experience has shown that such methods are not efficient when $\ker(W_{\mathrm{NN}}) \neq \{0\}$. The optimality conditions of this quadratic problem reduced to a linear complementarity problem with a semi-definite matrix. In that case, it is also possible to solve the problem with PSOR techniques with line-searches. Due to space constraints, we decided in this work to use the projected Gauss-Seidel (PGS) algorithm and the projected gradient algorithm of [Calamai and More, 1987] to solve the normal problem described. The projected gradient algorithm solved the following QP for a convex set C

$$\begin{cases} \min q(r) \coloneqq \dfrac{1}{2} r^{\top} W r + r^{\top} b \\ \text{s.t.} \quad r \in C \end{cases} \tag{127}$$

with the algorithm described in Algorithm 9.

The tangential problem,

$$\begin{cases} \min \dfrac{1}{2} r_{\mathrm{T}}^{\top} W_{\mathrm{TT}} r_{\mathrm{T}} + r_{\mathrm{T}}^{\top} \tilde{q}_{\mathrm{T}} \\ \text{s.t.} \quad r_{\mathrm{T}} \in D(\mu \tilde{r}_{\mathrm{N}}) \end{cases} \qquad \text{with } \tilde{q}_{\mathrm{T}} = q_{\mathrm{T}} + W_{\mathrm{TN}} r_{\mathrm{N},k+1}, \tag{128}$$

Algorithm 9 Projected gradient algorithm for QP (127)

Require: W, b that defines $q(r)$
Require: C a convex set
Require: r_0 initial values
Require: tol > 0 a tolerance value and $\mathsf{iter}_{\max} > 0$ the maximum number of iterations
Require: $\rho_0 > 0$, $l, \sigma \in (0,1)$
Require: $\mathsf{i}_{\mathsf{lsmax}}$ maximum number of line-search iterations
Ensure: r solution of Problem (127)
  $\mathsf{r_k} \leftarrow \mathsf{r}_0$; $\theta_0 \leftarrow \mathsf{q}(\mathsf{r}_0)$; $\mathsf{k} \leftarrow 0$
  while error $>$ tol and $\mathsf{k} < \mathsf{iter}_{\max}$ **do**
    Armijo like–search procedure
    $\mathsf{i_{ls}} \leftarrow 0$
    while criterion > 0 and $i_{ls} < i_{lsmax}$ **do**
      $\rho \leftarrow \rho_0{}^{\mathsf{i_{ls}}}$
      $\mathsf{r} \leftarrow \mathsf{P_C}(\mathsf{r_k} - \rho(\mathsf{M r_k} + \mathsf{b}))$
      $\theta \leftarrow \mathsf{q}(\mathsf{r})$
      $\mathsf{criterion} \leftarrow \theta - \theta_{\mathsf{k}} - \sigma(\mathsf{M r_k} + \mathsf{b})^\top(\mathsf{r} - \mathsf{r_k})$
      $i_{ls} \leftarrow i_{ls} + 1$
    end while
    $\mathsf{r_k} \leftarrow \mathsf{r}$; $\theta_{\mathsf{k}} \leftarrow \mathsf{q}(\mathsf{r})$;
    evaluate error.
  end while

is also a convex program but with a more complex structure since the constraints are quadratic one. There exists dedicated algorithm, as ([Dostál and Kozubek, 2012]), for QP with convex constraints. Earlier application of projected gradient and projected gradient techniques for the frictionless problem can also be found in [Barbosa et al., 1997], including a comparison with PSOR techniques.

In the report, we will use either a) a reformulation of the optimality conditions of this problem as a variational inequality and we apply the fixed point algorithm and the extra gradient algorithm of Section 4 or b) an adaptation of one of the splitting techniques detailed in Section 6. The algorithm is described in Algorithm 10. In Table 4, we detailed the algorithms we use in the present study.

7.2 Successive approximation method

The methods of successive approximations is a natural tool for the numerical realization of Problem FC. It is based on the Tresca approximation of the Coulomb cone as described in Section 3.4 and the work of the celebrated Czech school [Nečas et al., 1980, Haslinger, 1983, 1984, Haslinger et al., 1996]. Each iterative step is represented by an auxiliary contact problem with a given friction threshold described by quadratic program over a cylinder (93), that we recall there:

$$\begin{cases} \theta = h(r_{\mathrm{N}}) \\ \min \dfrac{1}{2} r^\top W r + r^\top q \\ \text{s.t.} \quad r \in C(\mu, \theta). \end{cases} \tag{129}$$

Algorithm 10 Panagiotopoulos decomposition algorithm for Problem FC

Require: $\mathsf{W}, \mathsf{q}, \mu$
Require: r_0 initial values
Require: $\mathsf{tol} > 0, \mathsf{tol}_{\mathsf{int}}$ tolerance values and $\mathsf{iter}_{\max} > 0$ the max number of iterations
Ensure: r, u solution of Problem FC
  $\mathsf{r}_k \leftarrow \mathsf{r}_0$; $\mathsf{k} \leftarrow 0$
  while error $>$ tol and $\mathsf{k} < \mathsf{iter}_{\max}$ **do**
    $\tilde{\mathsf{q}}_{\mathbf{N}} \leftarrow \mathsf{q}_{\mathbf{N}} + \mathsf{W}_{\mathbf{NT}}\mathsf{r}_{\mathbf{T},\mathsf{k}}$
    solve (126) for $r_{\mathbf{N},k+1}$ at accuracy $\mathsf{tol}_{\mathsf{int}}$
    $\tilde{\mathsf{q}}_{\mathbf{T}} \leftarrow \mathsf{q}_{\mathbf{T}} + \mathsf{W}_{\mathbf{TN}}\mathsf{r}_{\mathbf{N},\mathsf{k}+1}$
    solve (128) for $r_{\mathbf{T},k+1}$ at accuracy $\mathsf{tol}_{\mathsf{int}}$
    $\mathsf{k} \leftarrow \mathsf{k} + 1$
    evaluate error.
  end while
  $\mathsf{r} \leftarrow \mathsf{r}_k$
  $\mathsf{u} \leftarrow \mathsf{W}\mathsf{r} + \mathsf{q}$

The radius of the cylinder is then updated in an iterative procedure. The algorithm is described in Algorithm 11

Algorithm 11 Tresca approximation algorithm for Problem FC

Require: $\mathsf{W}, \mathsf{q}, \mu$
Require: r_0 initial values
Require: $\mathsf{tol} > 0, \mathsf{tol}_{\mathsf{int}}$ tolerance values and $\mathsf{iter}_{\max} > 0$ the max number of iterations
Ensure: r, u solution of Problem FC
  $\mathsf{r}_k \leftarrow \mathsf{r}_0$; $\mathsf{k} \leftarrow 0$
  while error $>$ tol and $\mathsf{k} < \mathsf{iter}_{\max}$ **do**
    $\theta \leftarrow h(r_{\mathbf{N},k})$
    solve (129) for $r_{\mathbf{N},k+1}, r_{\mathbf{T},k+1}$ at accuracy $\mathsf{tol}_{\mathsf{int}}$
    $\mathsf{k} \leftarrow \mathsf{k} + 1$
    evaluate error.
  end while
  $\mathsf{r} \leftarrow \mathsf{r}_k$
  $\mathsf{u} \leftarrow \mathsf{W}\mathsf{r} + \mathsf{q}$

In the literature, the successive approximation technique has been used in the bidimensional case in [Haslinger et al., 2002] & [Dostál et al., 2002] with improved and dedicated QP solvers over box-constraints. Two strategies are implemented: a) the classical Tresca iteration (called FPMI) and b) the Panagiotopoulos decomposition plus a Fixed point (called FPMII). They use a specific QP solver for box constraint [Dostál, 1997] that is an improvement of Moré–Toraldo method [Moré and Toraldo, 1991]. This technique has been directly extended in the three-dimensional case with a faceting of the cone in [Haslinger et al., 2004]. In the latter case, the problem is still a box constrained QP since it contains only polyhedral constraints. In [Haslinger et al., 2012] the authors propose a successive approximation technique in 3D with the special solver of [Kučera, 2007, 2008] which is itself an extension to disk constraints of the Polyak method (conjugate gradient with active set on the bounds constraint) and its improvements [Dostál, 1997,

Dostál and Schöberl, 2005]. Other improvements of the method may be found in [Dostál and Kučera, 2010] with a last improvement of the method in [Dostál and Kozubek, 2012]. All this work is summarized and details in [Dostál, 2016].

7.3 ACLM approach

In the convex SOCCP approach described in Section 3.4, we have to solve for a given value the following problem

$$\begin{cases} \min & \frac{1}{2} r^\top W r + r^\top (q+s) \\ \text{s.t.} & r \in K. \end{cases} \tag{130}$$

which is again a convex quadratic program over second–order cone. The approach listed above could again be used to solve this problem. In this work, we solve it by three different ways: a) an adaptation of one of the splitting techniques detailed in Section 6, b) using the projected gradient algorithm dedicated to convex QP described in Algorithm 9 or c) the fixed point algorithm and the extra gradient algorithm of Section 4 . The algorithm is described in Algorithm 12 and we detailed the algorithms we use in the present study in Table 4.

Algorithm 12 ACLM approximation algorithm for Problem FC

Require: $\mathsf{W}, \mathsf{q}, \mu$
Require: r_0 initial values
Require: $\mathsf{tol} > 0, \mathsf{tol}_{\mathsf{int}}$ tolerance values and $\mathsf{iter}_{\max} > 0$ the max number of iterations
Ensure: r, u solution of Problem FC
  $\mathsf{u}_0 \leftarrow \mathsf{W}\mathsf{r}_0 + \mathsf{q}$; $\mathsf{k} \leftarrow 0$
  while error $>$ tol and $\mathsf{k} < \mathsf{iter}_{\max}$ **do**
    $\mathsf{s} \leftarrow \mathsf{g}(\mathsf{u}_\mathsf{k})$
    solve (130) for r_{k+1} at accuracy $\mathsf{tol}_{\mathsf{int}}$
    $\mathsf{u}_{\mathsf{k}+1} \leftarrow \mathsf{W}\mathsf{r}_{\mathsf{k}+1} + \mathsf{q}$
    $\mathsf{k} \leftarrow \mathsf{k} + 1$
    evaluate error.
  end while
  $\mathsf{r} \leftarrow \mathsf{r}_\mathsf{k}$
  $\mathsf{u} \leftarrow \mathsf{u}_\mathsf{k}$

A nomenclature for the algorithms based on the optimisation approach is given in Table 4.

7.4 Convex relaxation and the SOCCP approach

Finally, we propose to compare the optimization based algorithm to a complete convex relaxation of the problem by solving the convex SOCCP (130) with $s = 0$. This procedure is very similar to the approach in [Tasora and Anitescu, 2009, Anitescu and Tasora, 2010, Tasora and Anitescu, 2011] where only the convex problem is solved.

7.5 Control of the tolerance of internal solvers $\mathsf{tol}_{\mathsf{int}}$ in optimization approach

In Algorithms 10, 11 and 12, an internal tolerance is used to control the accuracy of the internal solver. It is generally not useful to solve the internal problem at the accuracy of the global one. In the comparison study, we set the internal tolerance $\mathsf{tol}_{\mathsf{int}}$ to $\mathsf{error}/10$.

Name	Algo.	Additional informations
PANA-PGS-FP-VI-UPK	10	The normal problem is solved by a PGS algorithm and the tangent problem is solved with the FP-VI-UPK algorithm
PANA-PGS-EG-VI-UPK	10	The normal problem is solved by a PGS algorithm and the tangent problem is solved with the EG-VI-UPK algorithm
PANA-PGS-CONVEXQP-PG	10 &9	The normal problem is solved by a PGS algorithm and the tangent problem is solved with Algorithm 9
PANA-CONVEXQP-PG	10&9	Both normal and tangent problems are solved with Algorithm 9
TRESCA-NSGS-FP-VI-UPK	11	The problem 129 is solved with the FP-VI-UPK algorithm
TRESCA-FP-VI-UPK	11 & 1	The problem 129 is solved with the FP-VI-UPK algorithm
TRESCA-EG-VI-UPK	11 & 2	The problem 129 is solved with the EG-VI-UPK algorithm
TRESCA-CONVEXQP-PG	11 &9	The problem 129 is solved with Algorithm 9
ACLM-NSGS-FP-VI-UPK	12	The problem 130 is solved with the NSGS-FP-VI-UPK algorithm
ACLM-FP-VI-UPK	12& 1	The problem 130 is solved with the FP-VI-UPK algorithm
ACLM-EG-VI-UPK	12 & 2	The problem 130 is solved with the EG-VI-UPK algorithm
ACLM-CONVEXQP-PG	12	The problem 130 is solved with the Algorithm 9

Table 4: Naming convention for the optimization algorithms.

8 Comparison framework

In this section, we present our comparison framework. Especially, we specify how the performance is measured and how the performance profiles are built.

8.1 Measuring errors

A key parameter in the measurement of performance of the solver is the definition of the error. The absolute error is given by the norm of the natural map. A relative error is computed with respect to the norm of the vector q. More precisely, the error is given by

$$\text{error} = \frac{\|F_{\text{vi}}^{\mathsf{nat}}(r)\|}{\|q\|}. \tag{131}$$

assuming that $\|q\|$ is larger than the machine accuracy. If not, we may assume that $q = 0$ and a trivial solution can be computed. For all solvers, the error in (131) is compared to the required tolerance tol given by the user.

For some iterative solvers such as VI-FP, VI-EG, NSGS and PSOR, the computation of the error (131) at each iteration penalizes the performance of the solver: it amounts to computing a matrix-vector product, an operation that is more computationally expensive than one iteration of the solver. Hence, a cheaper error measurement is used inside the main loop in Algorithms 1, 2 and 7. This cheaper error measurement is given by

$$\text{error}_{\text{cheap}} = \frac{\|r_{k+1} - r_k\|}{\|r_k\|}. \tag{132}$$

The tolerance of solver is then self-adapted in the loop to meet the required tolerance based on the error given by (131).

8.2 Performance profiles

The concept of performance profiles was introduced in [Dolan and Moré, 2002] for bench-marking optimization solvers on a large set of problems. For a set P of n_p problems, and a set S of n_s solvers, we

define a performance criterion for a solver s, a problem p and a required precision tol by

$$t_{p,s} = \text{computing time required for } s \text{ to solve } p \text{ at precision tol}, \tag{133}$$

A performance ratio over all the solvers is defined by

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s},\ s \in S\}} \geqslant 1\,. \tag{134}$$

For $\tau \geqslant 1$, we define a distribution function ρ_s for the performance ratio for a solver s as

$$\rho_s(\tau) = \frac{1}{n_p}\text{card}\{p \in P,\ r_{p,s} \leqslant \tau\} \leqslant 1. \tag{135}$$

This distribution computes the number of problems p that are solved with a performance ratio below a given threshold τ. In other words, $\rho_s(\tau)$ represents the probability that the solver s has a performance ratio not larger than a factor τ of the best solver. It is worth noting that $\rho_s(1)$ represents the probability that the solver s beats the other solvers, and $\rho_s(\tau)$ characterizes the robustness of the method for large values of τ. To summarize: the higher ρ_s is, the better the method is. In the sequel, the term *performance profile* denotes a graph of the functions $\rho_s(\tau), \tau \geqslant 1$.

The computational time is used to measure performance in (133). Other criteria can be used, like the number of floating point operations (flops). It is a better measure of performance since it is independent of the computer. Unfortunately, it is usually difficult to measure in an automatic and robust way over various platforms. Whence, we stick with the computational time.

In our experiments, we decided to fix the required accuracy with the tolerance of each solver. Another performance criteria could also be used: for instance a timeout could be defined and the metric would be the error at that time. This a way to measure the ability of a solver to give an approximate solution within a prescribed time limit that may be interesting for real-time applications. Another way to measure performance may also be to divide the computational time by the number of contacts in order to judge of the ability of the solver to be scalable. For the sake of conciseness, this has not been done in this report.

8.3 Benchmarks presentation

To perform the comparison of the solvers on a fair basis, we use a large set of problems that comes from various applications. This collection is FCLib (Frictional Contact library) which is an open collection of problems in a hdf5 format described in [Acary et al., 2014][3]. In this work, we used the version `v1.0` for the comparisons that contains 2368 problems[4].

The test sets are illustrated in Figure 3 and details on each test are given in Table 5. All the problems have been generated thanks to the software codes LMGC90[5] and Siconos. In Table 5, the number of degrees of freedom n corresponds to the degrees of freedom of the system before its condensation (or reduction) to local variables. In other words, the number of rows of the matrix M and H in (1). The contact density c is the ratio of the number of contact unknowns over the number of degrees of freedom:

$$c = \frac{3n_c}{n} = \frac{m}{n}. \tag{136}$$

The coefficient c corresponds also to the ratio between the number of rows of H over its number of columns. If this number is larger than 1, the matrix H can not be full row rank and then the matrix W is also rank deficient. Whenever $m > n$, we can observe in Table 5 that this number c is a good

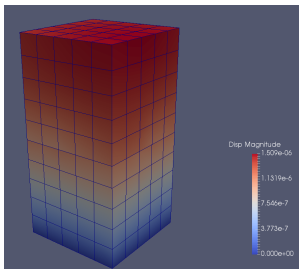
[3] More information can be found at https://frictionalcontactlibrary.github.io

[4] The whole collection of problems can be found at https://github.com/FrictionalContactLibrary/fclib-library

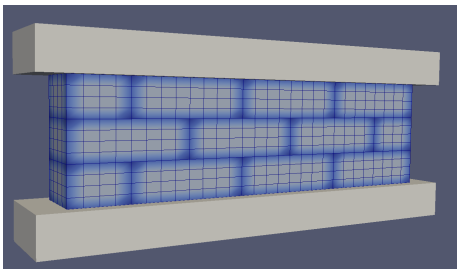
[5] https://git-xen.lmgc.univ-montp2.fr/lmgc90/lmgc90_user/wikis/home

approximation of the rank ratio of the matrix W in our applications. The estimation of the rank of matrix W shows that it is very close to the number of degrees of freedom of the system when $c > 1$. For $c \ggg 1$, the contact density is really high and the system suffers from hyperstaticity as we discussed in Section 2.3. In Table 5, we also give an estimation of the conditioning of the matrix W. When is was possible from a computational point of view, we perform a singular value decomposition (SVD) of the matrix W to estimate the spectral radius and then the conditioning by cutting the small eigenvalues. This process has two drawbacks. Firstly, the computation of the SVD decomposition can be really expensive for large dense matrices. Secondly, the value of the condition number of the matrix is very sensitive to the threshold for cutting off the small eigenvalues. This is the reason why we also use LSMR [Fong and Saunders, 2011] algorithm to give an better approximation of the condition number of rank deficient matrix.

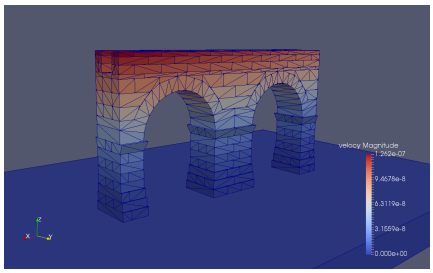
The four first tests in Table 5, Cubes_H8_2, Cubes_H8_5, Cubes_H8_20 and LowWall_FEM, are examples that involve flexible elastic bodies meshed by finite element methods. Due to a consistent choice of the space-discretization of the contact surfaces, the Delassus matrix W in that case is full rank. In the sequel, we will call these sets of examples the *flexible test sets*.



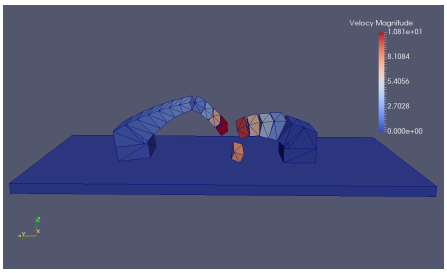
(a) Cubes_H8



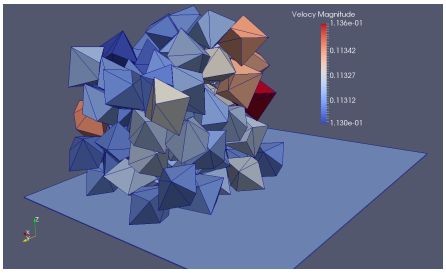
(b) LowWall_FEM



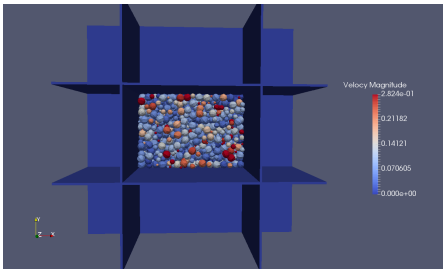
(c) Aqueduct_PR



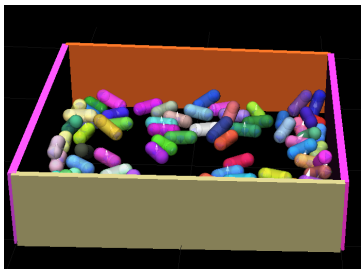
(d) Bridge_PR



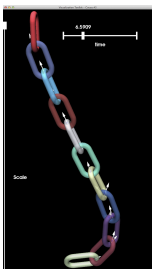
(e) 100_PR_Periobox



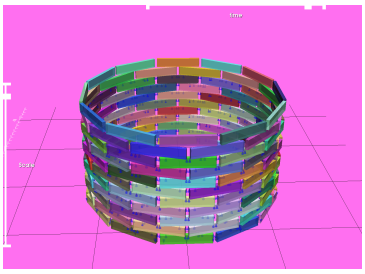
(f) 945_SP_Box_PL



(g) Capsules



(h) Chain



(i) KaplasTower

(j) BoxesStack

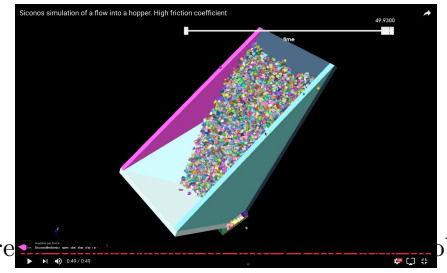


Figure oblems

(k) Chute_1000, Chute_4000, Chute_local_problems

Test set	code	friction coefficient μ	# of problems	# of d.o.f.	# of contacts	contact density c	rank ratio(W)	cond(W)	cond(W) LSMR	symmetry of W $\mu(\frac{\|W-W^T\|}{\|W\|})$
Cubes_H8_2	LMGC90	0.3	15	162	[3 : 5]	[0.02 : 0.09]	1	$[2.2.10^1 : 1.3.10^3]$	$[8.1.10^5 : 1.5.10^6]$	$3.2.10^{-4}$
Cubes_H8_5	LMGC90	0.3	50	1296	[17 : 36]	[0.02 : 0.09]	1	$[3.3.10^4 : 7.2.10^4]$	$[1.3.10^6 : 3.1.10^6]$	$4.2.10^{-4}$
Cubes_H8_20	LMGC90	0.3	50	55566	[361 : 388]	[0.019 : 0.021]	1	$[2.4.10^5 : 2.5.10^5]$	$[1.3.10^6 : 5.2.10^6]$	$5.2.10^{-5}$
LowWall_FEM	LMGC90	0.83	50	{7212}	[624 : 688]	[0.28 : 0.29]	1	–	$[9.3.10^2 : 5.0.10^5]$	$5.2.10^{-2}$
Aqueduct_PR	LMGC90	0.8	10	{1932}	[4337 : 4811]	[6.81 : 7.47]	[6.80 : 7.46]	$[4.7.10^7 : 3.4.10^8]$	$[6.7.10^1 : 1.5.10^2]$	$1.1.10^{-15}$
Bridge_PR	LMGC90	0.9	50	{138}	[70 : 108]	[1.5 : 2.3]	[2.27 : 2.45]	$[8.3.10^4 : 1.1.10^5]$	$[1.9.10^3 : 2.6.10^4]$	$5.8.10^{-18}$
100_PR_Periobox	LMGC90	0.8	106	{606}	[14 : 578]	[0.2 : 3]	[1.76 : 3.215]	$[4.3.10^2 : 1.0.10^6]$	$[6.3.10^5 : 3.5.10^6]$	$8.8.10^{-20}$
945_SP_Box_PL	LMGC90	0.8	60	{5700}	[2322 : 5037]	[1.22 : 2.65]	[1.0 : 2.66]	$[2.2.10^4 : 4.4.10^5]$	$[2.9.10^1 : 9.2.10^2]$	$1.3.10^{-10}$
Capsules	Siconos	0.7	249	[96:600]	[17 : 304]	[0.53 : 1.52]	[1.08 : 1.55]	–	$[4.8 : 1.6.10^2]$	$3.3.10^{-02}$
Chain	Siconos	0.3	242	{60}	[8 : 28]	[0.5 : 1.3]	[1.05 : 1.6]	$[7.4.10^4 : 4.0.10^9]$	$[1.5.10^1 : 4.7.10^5]$	$3.7.10^{-02}$
KaplasTower	Siconos	0.7	201	[72 : 792]	[48 : 933]	[3.0 : 3.6]	[2.0 : 3.53]	[67 : 2174]	[8 : 67]	$5.4.10^{-08}$
BoxesStack	Siconos	0.7	255	[6 : 300]	[1 : 200]	[1.86 : 2.00]	[1.875 : 2.0]	$[3.8.10^4 : 2.5.10^7]$	$[9.0 : 5.4.10^3]$	$2.23.10^{-14}$
Chute_1000	Siconos	1.0	156	[276 : 5508]	[74 : 5056]	[0.69 : 2.95]	[1.0 : 2.95]	$[2.1.10^1 : 1.9.10^3]$	$6.6.10^{-02}$	
Chute_4000	Siconos	1.0	140	[1280 : 20034]	[459 : 19795]	[2.51 : 3.06]	–	$[5.5.10^1 : 9.0.10^3]$	$8.9.10^{-14}$
Chute_local_problems	Siconos	1.0	834	3	1	1	1	[1.04 : 4.66]	$[2.6 : 2.6.10^1]$	$1.76.10^{-09}$

8.4 Software & implementation details

All the solvers that are used in this report are implemented in standard C99 in the component of the open source software Siconos called numerics. The aim of Siconos is to provide a common platform for the modeling, simulation, analysis and control of general nonsmooth dynamical systems[6]. The linear algebra operations are based on BLAS/LAPACK. The algorithms VI-FP, VI-EG, NSGS and PSOR use the sparse block structure of the Delassus matrix W. The NSN solvers relies on a standard sparse implementation given by csparse[7]. We solve linear systems with the LU factorization method embedded in csparse. The simulations are performed on the University of Grenoble-Alpes cluster CIMENT[8].

8.5 Simulation campaign

The simulation campaign is described in Table 6. For some test sets, two simulation runs have been performed with different precisions and prescribed time limits. A trade-off between the time limit and the precision has been chosen such that all the problems of the test sets are solved by at least one solver. In Sections 9 and 10, we report the results for the simulation campaign, which includes more that 27000 runs. Given this wealth of data, we do not report in this report, profiles when a family of solvers fails to solve the instances[9].

9 Comparison of methods by family

In this section, we perform a comparison of the solvers by family. The goal is to study the influence of the various parameters and possible strategies on the performance of the solvers.

9.1 Numerical methods for VI: FP-DS, FP-VI-$\star$ and FP-EG-$\star$

In Figure 4, we compare the different VI numerical solvers described in Section 4. Except for the FP-DS solver, the solvers FP-VI-$\star$ and FP-EG-$\star$ are very robust. Nevertheless, they are quite slow to converge in practice for large problems and/or with tight tolerances. Only the test sets for which the solvers have reached the precision before the prescribed time limit are presented. For that reason, the results for the test sets LowWall_FEM, LowWall_FEM II, Cubes_H8, Bridge_PR, AqueducPR, 945_SP_Box_PL, BoxesStack, Chute_4000 and Chute_1000 are not depicted. The main conclusions are as follows:

1. The solver FP-DS suffers from robustness problems and a lot of divergence has been observed. This is mainly due to the fact that we set a priori the ρ parameter in Algorithm 1 to a fixed value equal to 1, independently of the problem.

2. The solvers FP-VI-$\star$ and FP-EG-$\star$ are really robust but slow. They are able to solve all the problems but they require a lot of time. We did not observe divergence issues on all the test sets for these solvers. Comparing to FP-DS, the self-adaptive rule for sizing the paramater ρ_k is of utmost importance for the robustness and the convergence rate.

[6]More information on the software is available at http://siconos.gforge.inria.fr and the software can be downloaded at https://github.com/siconos/siconos

[7]http://people.sc.fsu.edu/ jburkardt/c_src/csparse/csparse.html

[8]https://ciment-grid.ujf-grenoble.fr/

[9]Nevertheless, the reader can have access to the complete list of performance profiles at https://github.com/siconos/faf/blob/master/TeX/Full-test/full-test_current.pdf

Test set	precision	prescribed time limit (s)	mean performance of the fastest solver $\mu\{\min\{t_{p,s}, s \in S\}\}$	std. deviation performance of the fastest solver $\sigma(\min\{t_{p,s}, s \in S\})$	mean performance of the fastest solver by contact $\mu\{\min\{t_{p,s}/n_{c,p}, s \in S\}\}$	std. deviation performance of the fastest solver by contact $\sigma(\min\{t_{p,s}/n_{c,p}, s \in S\})$	# of unsolved problems
Cubes_H8_⋆	10^{-08}	100	1.73	2.13	4.83^{-03}	5.78^{-03}	0
Cubes_H8_⋆ II	10^{-04}	100	0.92	1.06	2.66^{-03}	2.83^{-03}	0
LowWall_FEM	10^{-08}	400	13.1	3.50	1.91^{-02}	5.09^{-03}	0
LowWall_FEM II	10^{-04}	400	14.8	2.85	2.16^{-02}	4.54^{-03}	0
Aqueduct_PR	10^{-04}	200	5.80	6.36	4.90^{-04}	3.03^{-04}	0
Bridge_PR	10^{-08}	400	10.3	12.9	1.23^{-01}	2.88^{-01}	0
Bridge_PR II	10^{-04}	100	0.048	0.038	1.30^{-03}	1.42^{-03}	0
100_PR_Periobox	10^{-04}	100	0.064	0.062	1.56^{-04}	1.22^{-04}	0
945_SP_Box_PL	10^{-04}	100	3.20	1.71	6.45^{-04}	3.36^{-04}	0
Capsules	10^{-08}	50	$1.46.10^{-02}$	$1.74.10^{-02}$	5.67^{-05}	6.26^{-05}	0
Chain	10^{-08}	50	$6.19.10^{-04}$	$3.68.10^{-04}$	$3.15.10^{-05}$	$1.46.10^{-05}$	0
KaplasTower	10^{-08}	200	$1.27.10^{-01}$	$3.75.10^{-01}$	$1.84.10^{-04}$	$4.57.10^{-04}$	0
KaplasTower II	10^{-04}	100	$2.84.10^{-02}$	$1.51.10^{-01}$	$3.39.10^{-05}$	$1.84.10^{-04}$	0
BoxesStack	10^{-08}	100	$3.42.10^{-02}$	$8.87.10^{-02}$	$3.24.10^{-04}$	$9.77.10^{-04}$	0
Chute_1000	10^{-04}	200	2.62	3.06	6.76^{-04}	6.58^{-04}	0
Chute_4000	10^{-04}	200	10.52	7.88	5.71^{-04}	4.07^{-04}	0
Chute_local_problems	10^{-08}	10	$1.80.10^{-04}$	$1.57.10^{-05}$	$1.80.10^{-04}$	$1.57.10^{-05}$	0

Table 6: Parameters of the simulation campaign

3. Except for the test set KaplasTower II, the FP-EG-⋆ performs better than FP-VI-⋆. Otherwise, the performance are quite similar since we plot the performance for a quite narrow range of values of $\tau \in [1, 5]$

4. The difference between the adaptive strategy for sizing ρ_k, UPK and UPTS, is negligible in all the test sets. Therefore, the choice of the update rule is not really important.

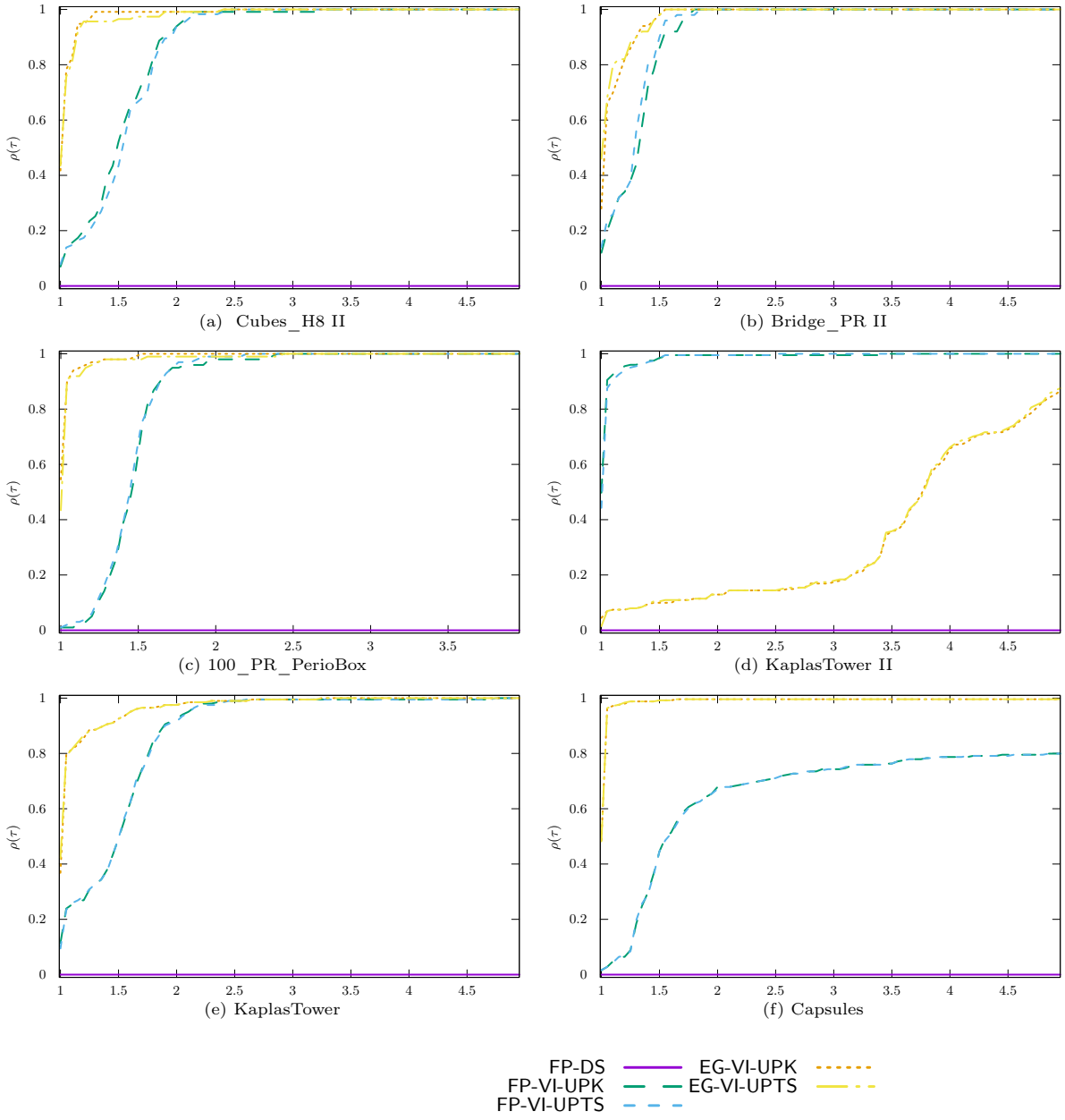


Figure 4: Comparison of numerical methods FP-DS, FP-VI-⋆ and FP-EG-⋆

9.2 Splitting based algorithms: NSGS-⋆ and PSOR-⋆

In this section, we compare the family of solvers based on splitting and relaxation techniques described in Section 6.1. Firstly, we start by comparing the choice of the local solvers in NSGS-⋆ and then the effect of the local tolerance $\mathsf{tol}_{\mathsf{local}}$. Secondly, we study the influence of the order of the contact list. Finally, we study the effect of the relaxation parameter ω in PSOR-⋆ solvers.

Influence of the local solver in NSGS-⋆ algorithms In Figure 5, we report the performance profiles of the NSGS-⋆ for the different local solvers. The main conclusions are:

1. When the prescribed time limit is sufficiently large and the tolerance is low (10^{-4}), we observe that the NSGS-⋆ solvers are robust. Indeed, we are able to find a local solver for each test sets that is able to give a solution at the required accuracy. Nevertheless, there is no universal efficient local solver that outperforms the other ones.

2. When the tolerance is equal to 10^{-8}, the NSGS-⋆ solvers have some difficulties to reach convergence for all the problems within the prescribed time limit. This is the case for the test sets LowWall_FEM, Cubes_H8, Bridge_PR, Chain, Capsules and BoxesStack. Generally, the convergence is so slow that it is difficult to reach tight tolerance with a reasonable time limit.

3. Except for the test sets KaplasTower II and BoxesStack, the solver NSGS-EXACT behaves poorly. This is mainly due to the fact that the local solver is not robust to find a solution when the unknowns are far from the global solution for all the other contacts. This behavior was already reported in [Daviet et al., 2011] where another solver based on a nonsmooth Newton technique is used when the exact solution is not satisfactory.

4. The NSGS-FP-DS-One solver is most efficient on the test sets Bridge_PR II, KaplasTower II, Chain and BoxesStack. In these tests sets, a part of the problems seems easier to solve and the NSGS-FP-DS-One solver seems sufficient to get a global convergence. Nevertheless, this local solver seems slow or suffers from robustness issues for other test sets.

5. On the flexible test sets, Cubes_H8_⋆, LowWall_FEM and the rigid test sets 945_SP_Box_PL and Chute_4000, the best solver is NSGS-FP-VI-UPK for a relatively low required tolerance ($\mathsf{tol}_{\mathsf{local}} = 10^{-06}$). For these test sets, an approximate solution of the single contact problems seems sufficient to ensure an efficient convergence towards the solution without entailing robustness.

6. On the test sets 100_PR_PerioBox, KaplasTower, Chain, Capsules, the solver NSGS-NSN-⋆ are the best solvers and behave very well on Bridge_PR II. It seems that when a tight accuracy is required, the solvers NSGS-NSN-⋆ are useful and helps with a tight local tolerance to speed-up the convergence.

7. For the Chute_1000, Chute_4000 test sets, we observe large differences between the local formulations of the nonsmooth equations for the Newton solvers (NSGS-NSN-AC or NSGS-NSN-JM). The solver NSGS-NSN-JM is the best solver and really better than NSGS-NSN-AC although their theoretical formulation are very close. These two test sets are characterized by difficult local problems where the Delassus matrix W is unsymmetric with large extra-diagonal terms.

8. For almost all the tests, the line–search procedures slow down the solvers without increasing the robustness. The only test sets where it has a positive outcome is Chute_4000 where the NSGS-AC solver fails to get a solution and the line–search seems to stabilize the algorithm.

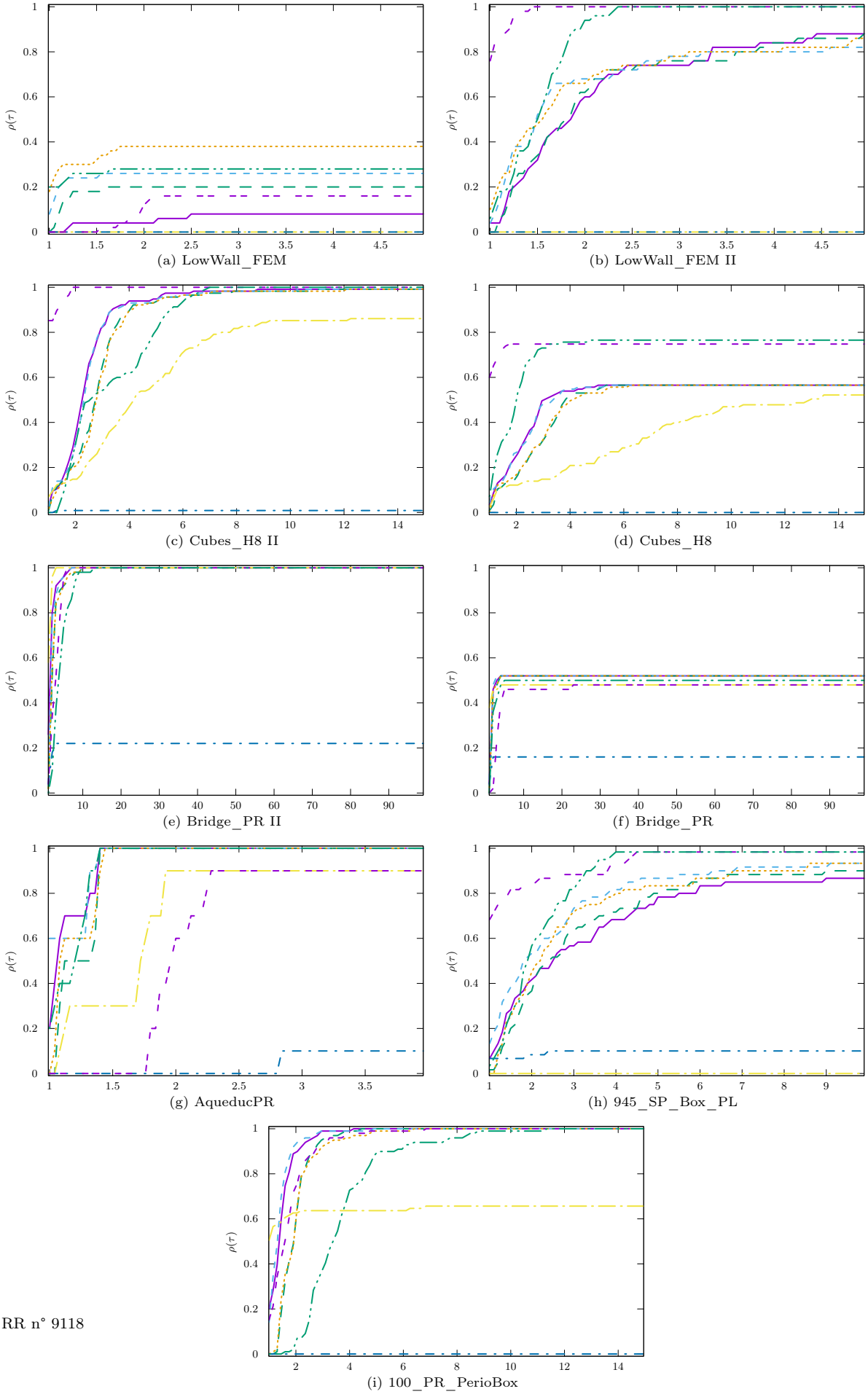


Figure 5: Influence of the local solver in NSGS-⋆ algorithms.

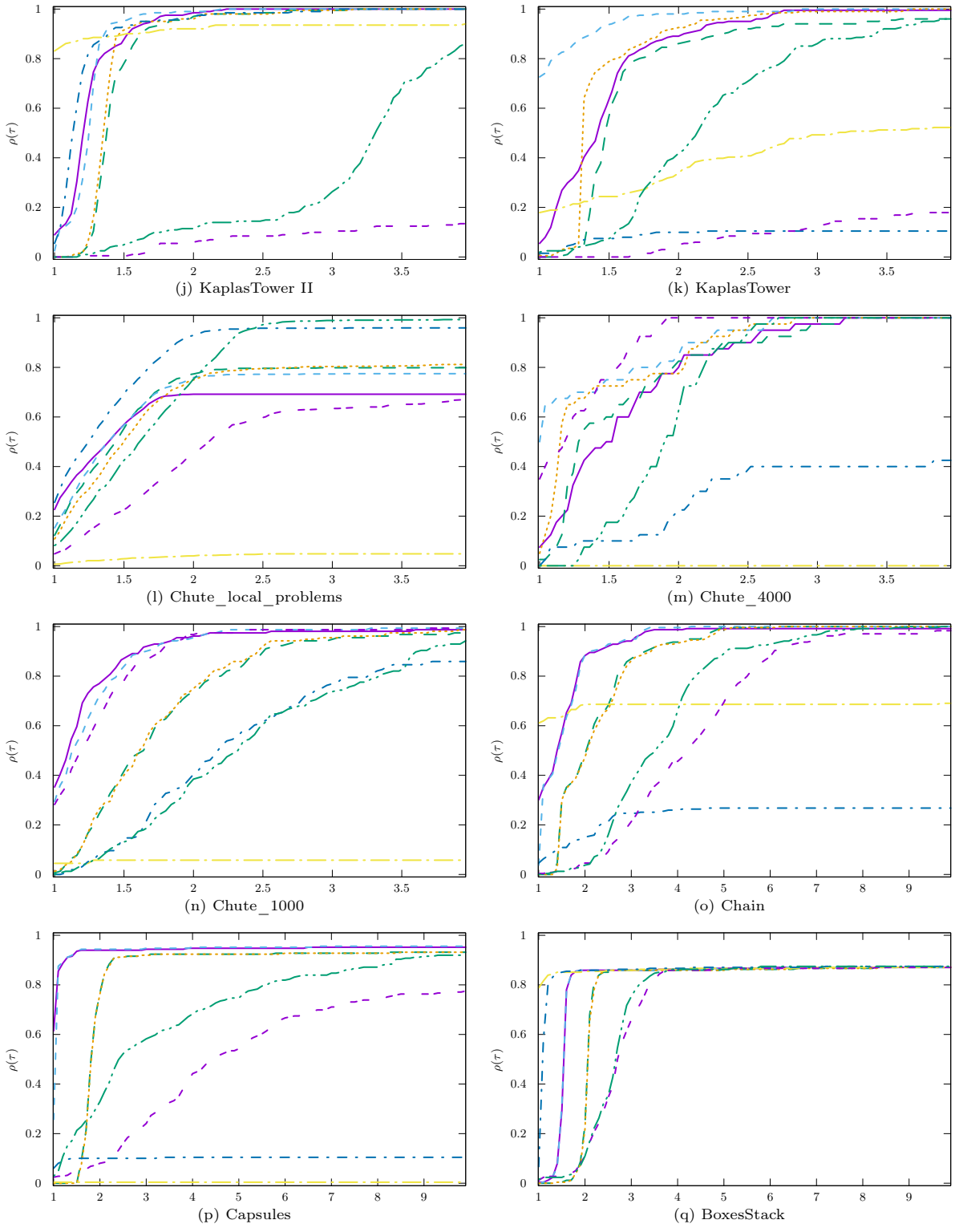


Figure 5: Influence of the local solver in NSGS-⋆ algorithms (continued).

Influence of the tolerance of the local solver $\mathsf{tol}_{\mathsf{local}}$ in NSGS-FP-VI-UPK algorithms In this paragraph, the tolerance of the local solver $\mathsf{tol}_{\mathsf{local}}$ is varied and its effect on the global convergence of the solver is reported. In Figure 6, we report the performance profiles of NSGS-FP-VI-UPK algorithms for the $\mathsf{tol}_{\mathsf{local}}$ in the range $[10^{-04}, 10^{-16}]$. We also report the efficiency of the adaptive strategy for sizing the value of the local tolerance (see Section 6.3). The main observations are:

1. For the test sets that are quickly solved (see Table 6), such as Capsules a tight tolerance on the local solver 10^{-16} improves the efficiency of the NSGS-FP-VI-UPK solver. Similar results are obtained for BoxesStack, Chain, KaplasTower and KaplasTower II; they are not depicted.

2. For the other problems that are harder to solve, that is, when we expect more iterations of the NSGS-FP-VI-UPK solver, the adaptive rule, or a tight local tolerance is better.

From these results, it is quite difficult to guess in advance the internal dynamics of the solver. By internal dynamics, we mean the propagation in the algorithm of the error and the values of the unknowns, between the local problem solvers and the global loop over contacts. Note that the range of τ that we used in the graph is quite small, so the difference in performance between the solvers is not crucial.

Influence of the tolerance of the local solver $\mathsf{tol}_{\mathsf{local}}$ in NSGS-AC-GP algorithms. In Figure 7, we report the performance profiles of NSGS-AC-GP algorithms for the $\mathsf{tol}_{\mathsf{local}}$ in the range $[10^{-04}, 10^{-16}]$. We also test the adaptive strategy for the local tolerance. Except for the test set Chute_local_problems, the main observation is that the local tolerance does not noticeably change the convergence of the solver. For the test set Chute_local_problems, there is no internal dynamics of the main loop of the NSGS since there is only one contact. It is therefore reasonable to see that the adaptive strategy performs better than the other.

Influence of the choice of the parameters $\rho_{\mathrm{N}}, \rho_{\mathrm{T}}$ in the local solver of the NSGS-AC algorithms In Figure 8, we evaluate the influence of the choice of the parameters $\rho_{\mathrm{N}}, \rho_{\mathrm{T}}$ on the convergence of the solver. The main conclusion are:

1. For the test sets 945_SP_Box_PL, 100_PR_PerioBox, KaplasTower II, KaplasTower, Chute_local_problems, Chute_4000, Chute_1000, Capsules, a fixed value of $\rho_{\mathrm{N}} = \rho_{\mathrm{T}} = 1$ has a dramatic effect on the convergence of the algorithm. The scaling of ρ is of utmost importance for the efficiency and the robustness of the solver. Note that the rule (112) that takes into account the condition number of the local Delassus matrix W deteriorates the performance for Chute_4000, Chute_1000. In these problems, the local matrix is unsymmetric with large extra-diagonal terms due to large gyroscopic effects.

2. For the other tests, the choice of $\rho_{\mathrm{N}}, \rho_{\mathrm{T}}$ does not really change the results such as LowWall_FEM II, mainly due the fact that the order of magnitude of the chosen ρ with the rules (110), (111) or (112) is in $[10^{-01}, 1]$. Cubes_H8 II, Cubes_H8, Bridge_PR II, Bridge_PR, 100_PR_PerioBox, Chain, BoxesStack and AqueducPR are not displayed since the results are similar.

One of the conclusions of this study is as follows: the rules (110), (111) improve a lot some simulations without increasing the computational cost for the others. Therefore, it is strongly advised to use them. Some further theoretical studies are needed to understand the effect of ρ on the convergence. In particular, the rule (112) is usually better, but sometimes completely destroys the convergence.

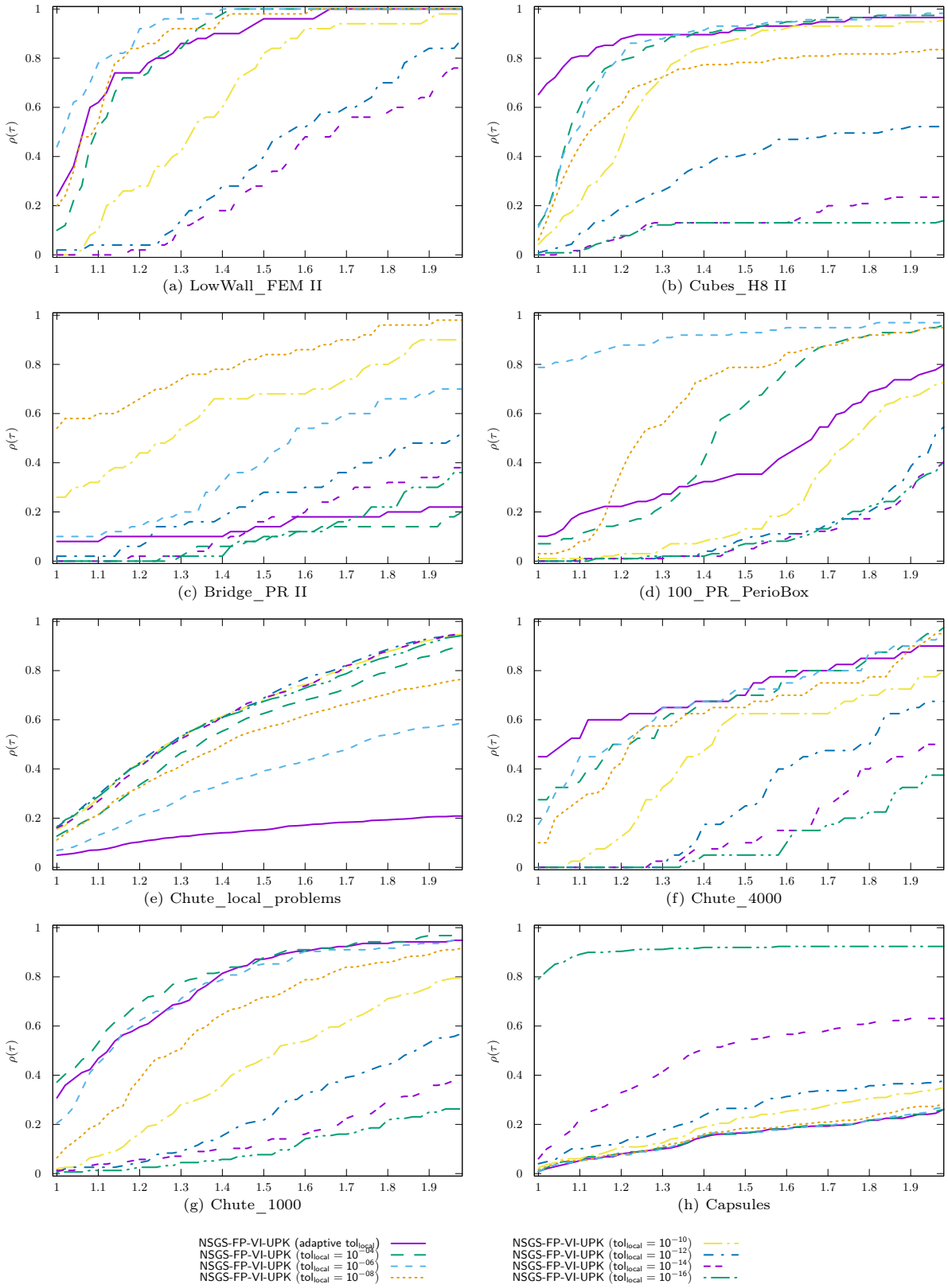


Figure 6: Influence of the tolerance of the local solver $\mathsf{tol}_{\mathsf{local}}$ in NSGS-FP-VI-UPK algorithms.

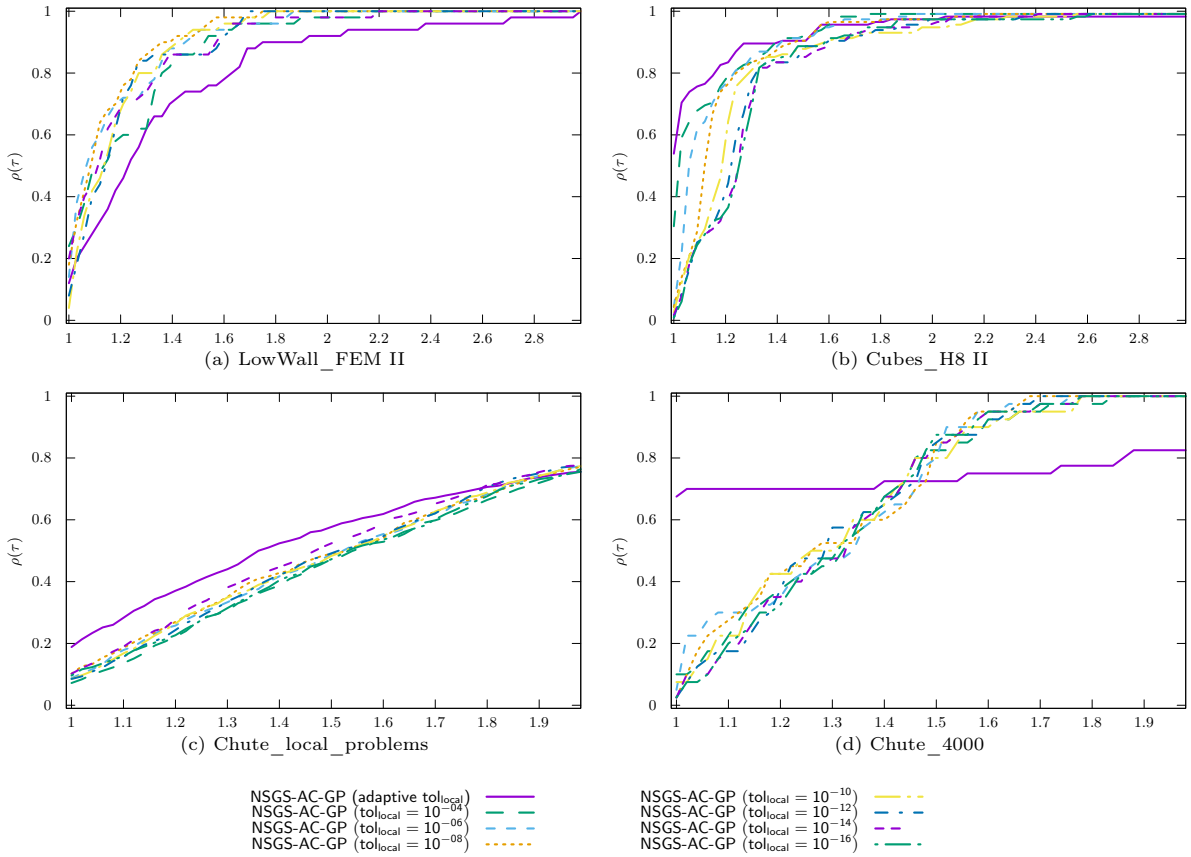


Figure 7: Influence of the tolerance of the local solver $\mathsf{tol}_{\mathsf{local}}$ in NSGS-FP-NSN-AC-GP algorithms.

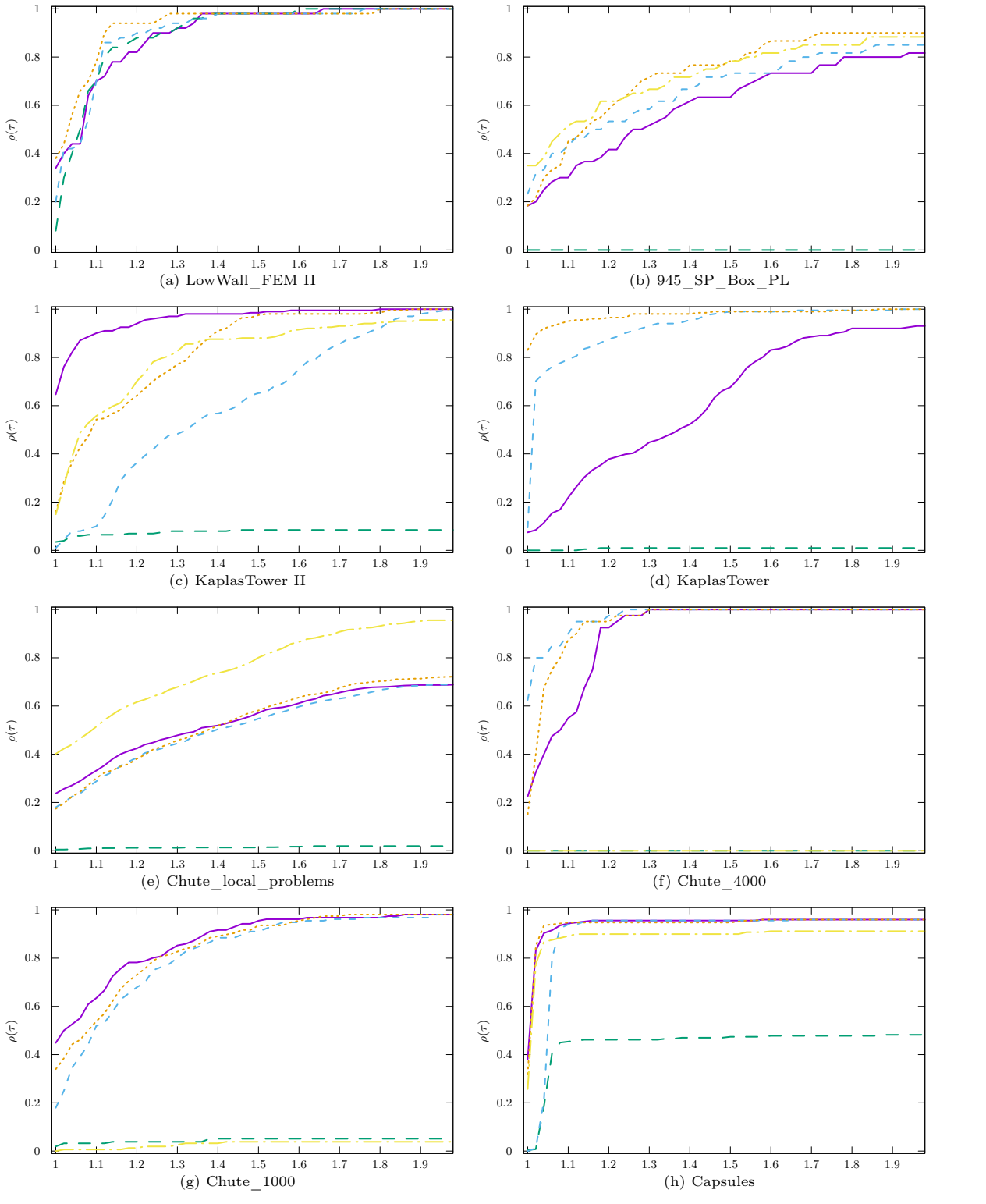


Figure 8: Influence of the choice of the parameters ρ_N, ρ_T in the local solver of the NSGS-AC algorithms

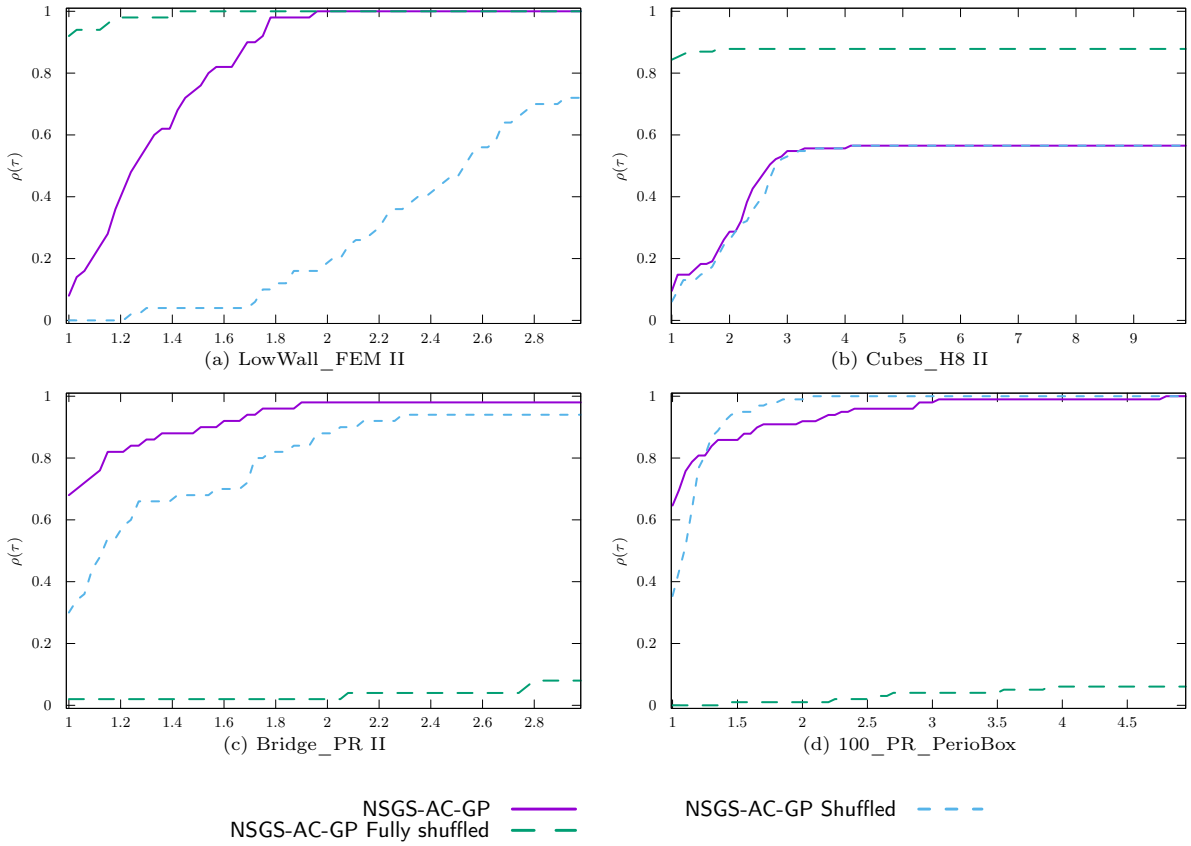


Figure 9: Influence of the contacts order in NSGS algorithms.

Influence of the order of contacts in NSGS algorithms In this section, we study the influence of the contact order within the loop of the NSGS-AC-GP solver. We reproduce in Figure 9 the result of the solvers with the original contact list of the problem (NSGS-AC-GP) and with two other ways of iterating over the contacts. The solver NSGS-AC-GP Shuffled corresponds to a single randomization of the list of contacts at the beginning of the algorithm. In the solver NSGS-AC-GP Fully shuffled, the list is shuffled at each iteration. The following observations can be made:

1. The solver NSGS-AC-GP Fully shuffled performs really better on the flexible test sets (Cubes_H8_$\star$, LowWall_FEM).

2. For the rigid test sets, we reproduce here only the test set 100_PR_PerioBox because the other test sets behave similarly. The NSGS-AC-GP Fully shuffled has a really bad influence on the convergence of the solver. It seems that it modifies the internal dynamics of the solver in a way that the rate of convergence is really decreased.

Comparison of PSOR algorithm with respect to the relaxation parameter ω In Figure 10, the relaxation parameter ω is varied ranging in $[0.5, 1.8]$. Two conclusions can be drawn:

1. For the flexible tests Cubes_H8_$\star$ and LowWall_FEM, the efficiency of the solver is really improved as we decreased the value of ω. Moreover, this is done without destroying the robustness of the solver.

2. For the rigid tests, the effect of the relaxation is not so clear. For values of ω greater than 1.0, the efficiency is improved but the robustness deteriorates. We observe the contrary for the ω less than 1.0. Note in particular that, for the test sets Chute_1000 and Chute_4000, the convergence is completely destroyed for $\omega = 1.8$.

To conclude, it is difficult to advice to use PSOR algorithm with $\omega \neq 1$. If it accelerates drastically the rate of convergence of the algorithm for some problems, but it deteriorates the convergence for others. Further studies would be needed to design self–adaptive schemes for sizing ω.

9.3 Comparison of NSN-$\star$ algorithms

In this section, the nonsmooth Newton methods are compared. The performance profiles are depicted in Figure 11 for the test sets for which the NSN-$\star$ are able to solve at least 10% of the problems. The main conclusions are as follows:

1. For the flexible tests Cubes_H8_$\star$ and LowWall_FEM, most of the Newton methods succeed to solve the problems within the prescribed time limit. The solver NSN-AC-HYBRID appears to be the best solver. The effect of computing an initial guess with a robust method such as EG-VI-UPK improves the convergence. In practice, we observe that the computation allows one to determine roughly the set of closed and sliding contacts and it helps a lot the convergence of the Newton solvers. The solvers without a line–search procedure perform also better than those with a line–search procedure which seems to slow down the convergence without improving the robustness. For the different formulations, the NSN-AC and NSN-JM give equivalent results and are better than the NSN-NM solver which is in turns better than the NSN-FB solver. Note that the Goldstein–Price line search is usually better than the Armijo despite the fact that the merit function is not necessarily smooth. Finally, we note that NSN-FB and NSN-FB-A are really the slowest solvers on these flexible examples.

2. For the rigid test sets with a high value of the rank ratio or the contact density c (see Table 5), the Newton methods fail to converge and a lot of divergence issues have been noted in practice. This is the case for the test sets Bridge_PR II, Bridge_PR, AqueducPR, 945_SP_Box_PL, 100_PR_PerioBox that are not depicted in Figure 11.

3. For the rigid test sets with a low value of the rank ratio or the contact density c less than 1 such as Chute_1000 and Chain, we observe that the Newton methods are able to solve some problems. We note also that in the Chain test set, the use of a fixed value of ρ is penalizing a lot the convergence of the solver. Contrary to flexible test sets, the use of a line–search procedure helps to get a better robustness of the solver. This is particularly true for NSN-NM-GP.

4. Finally, for the test sets KaplasTower and Capsules, the NSN-FB-GP is able to solve more than 80% of the tests in a very efficient way. Some further studies would be needed to understand why this specific solver performs really better than the others.

As a general conclusion, the success of the NSN-$\star$ algorithms is conditioned by the rank of the Delassus matrix W, and then, by the contact density value c. For full rank matrix W, the solvers are robust and efficient. For values of c not larger than 1, the methods are able to find a solution with a tight accuracy.

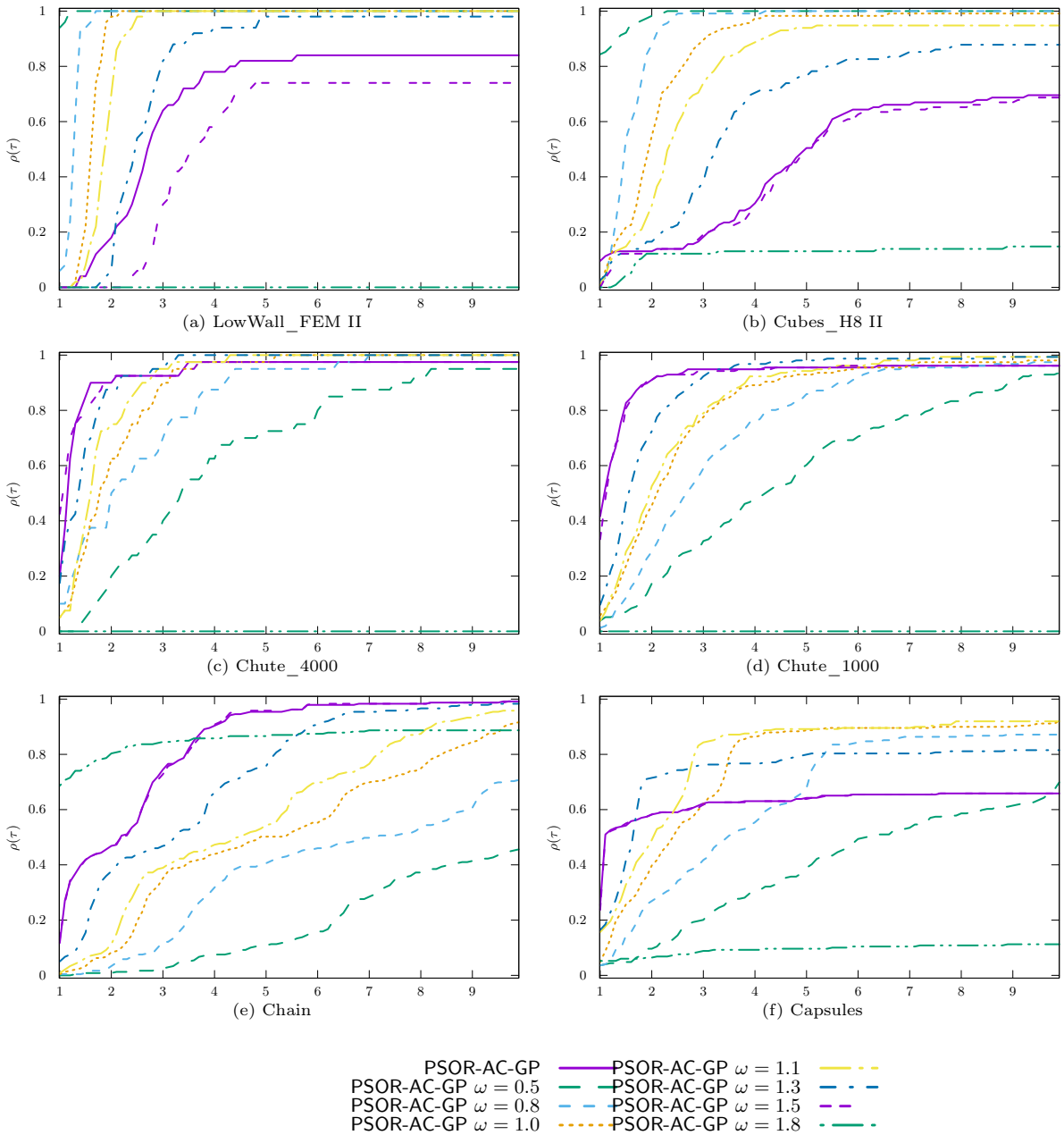


Figure 10: Effect of relation coefficient ω in PSOR-AC-GP algorithm.

For larger values of c and larger rank ratio, the nonsmooth Newton methods are not robust and generally diverge.

9.4 Comparison of the proximal point algorithm PPA-NSN-⋆ and PPA-NSGS-⋆ algorithms

In Figure 12, we compare the proximal point approach with various internal solvers based on nonsmooth Newton methods NSN-⋆. The main observations are:

1. For the flexible test sets (see for an illustration the test set LowWall_FEM II), for which the nonsmooth Newton solvers work pretty well, the use of a proximal point algorithm has no interest since it slows down the convergence of the algorithm by performing a first iteration with a given, and possibly large, value of the parameter α.

2. For the test sets KaplasTower, Chute_1000, Chain, Capsules and BoxesStack, the proximal point approach improves greatly the efficiency of the NSN-AC-GP solver and often also improves its reliability (see for comparison Figure 11). Clearly, the regularization introduced in the proximal point algorithm increases the rank of the matrix W and it has a strong effect on the convergence of the nonsmooth Newton methods.

3. The efficiency of the proximal point algorithm strongly depends on the internal solver.

4. The strategy for updating the regularization parameter α plays also an important role. Quite surprisingly, for the Bridge_PR test set, the adaptive rule that does not take into account the current error is really efficient and allows us to get a robust and efficient solver with respect to the others. Unfortunately, there is no updating rule for the parameter α that works for all test sets.

In Figure 13, we compare the NSGS-AC solver when it is used directly or inside the proximal point algorithm. On most of the test sets such as KaplasTower, a direct application of the NSGS-AC solver is already efficient and its embedding into a proximal point algorithm does not bring any improvements. Nevertheless, we can see in Figure 13 that the proximal point algorithm improves the robustness and the efficiency for the test sets 945_SP_Box_PL and Capsules has been improved.

9.5 Comparison of optimization-based algorithms PANA-⋆, TRESCA-⋆ and ACLM-⋆

In Figure 14, we compare the algorithms based on the optimization approach presented in Section 7. The pure convex relaxation SOCLCP-NSGS-PLI method has been added to understand the effect of the nonconvexity of the problems on the efficiency and robustness of the solvers. The main conclusions are:

1. The pure convex relaxation in SOCLCP-NSGS-PLI simplifies drastically the problems in the test sets LowWall_FEM II, AqueducPR, KaplasTower, BoxesStack and is slightly better in Bridge_PR II, 100_PR_PerioBox, KaplasTower II test sets. Especially, we note that if we want to reach a better accuracy as in the KaplasTower test set, the convex relaxation helps a lot, but this conclusion cannot be made in the test set Bridge_PR. Let us also note that the convex relaxation does not help a lot in the test sets Cubes_H8, Bridge_PR, Chute_1000, Chute_4000 and Capsules. One of the conclusion may be that the nonconvexity of the problem is not the only difficulty in such problems. Using a convex relaxation is not sufficient to solve all the problems.

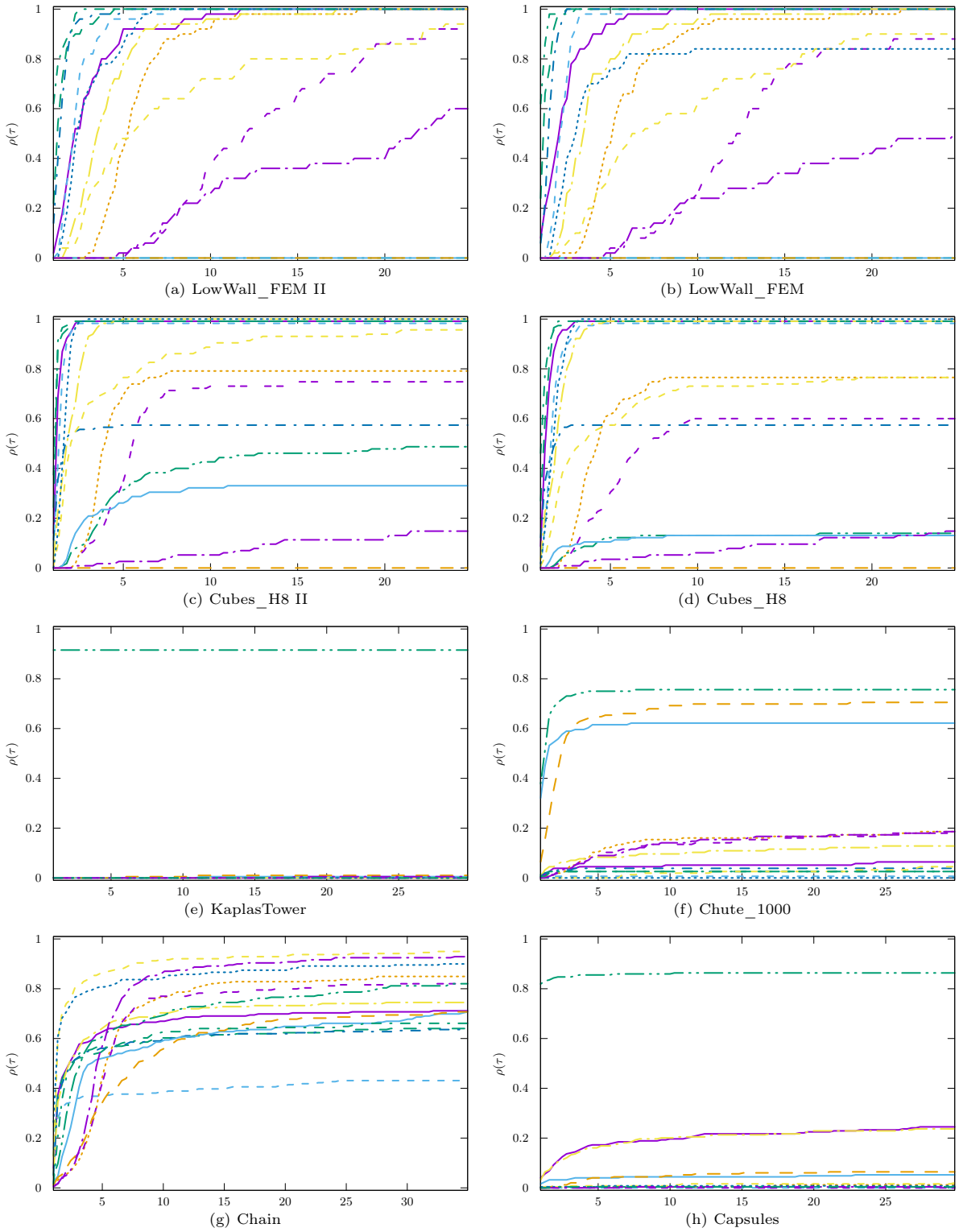


Figure 11: Comparison of NSN-$\star$ algorithms.

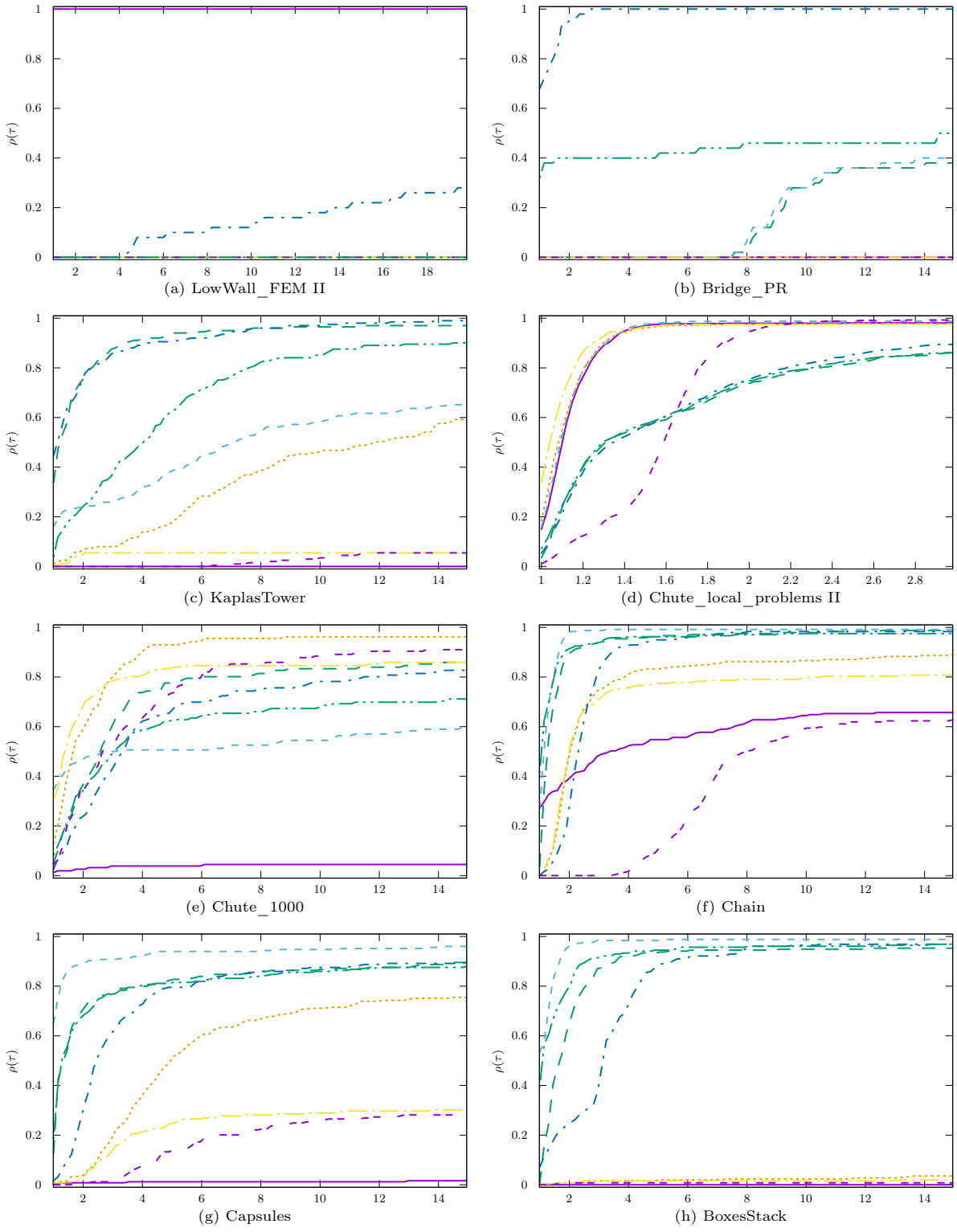


Figure 12: Comparison of internal solvers in PPA-NSN-⋆ algorithms.

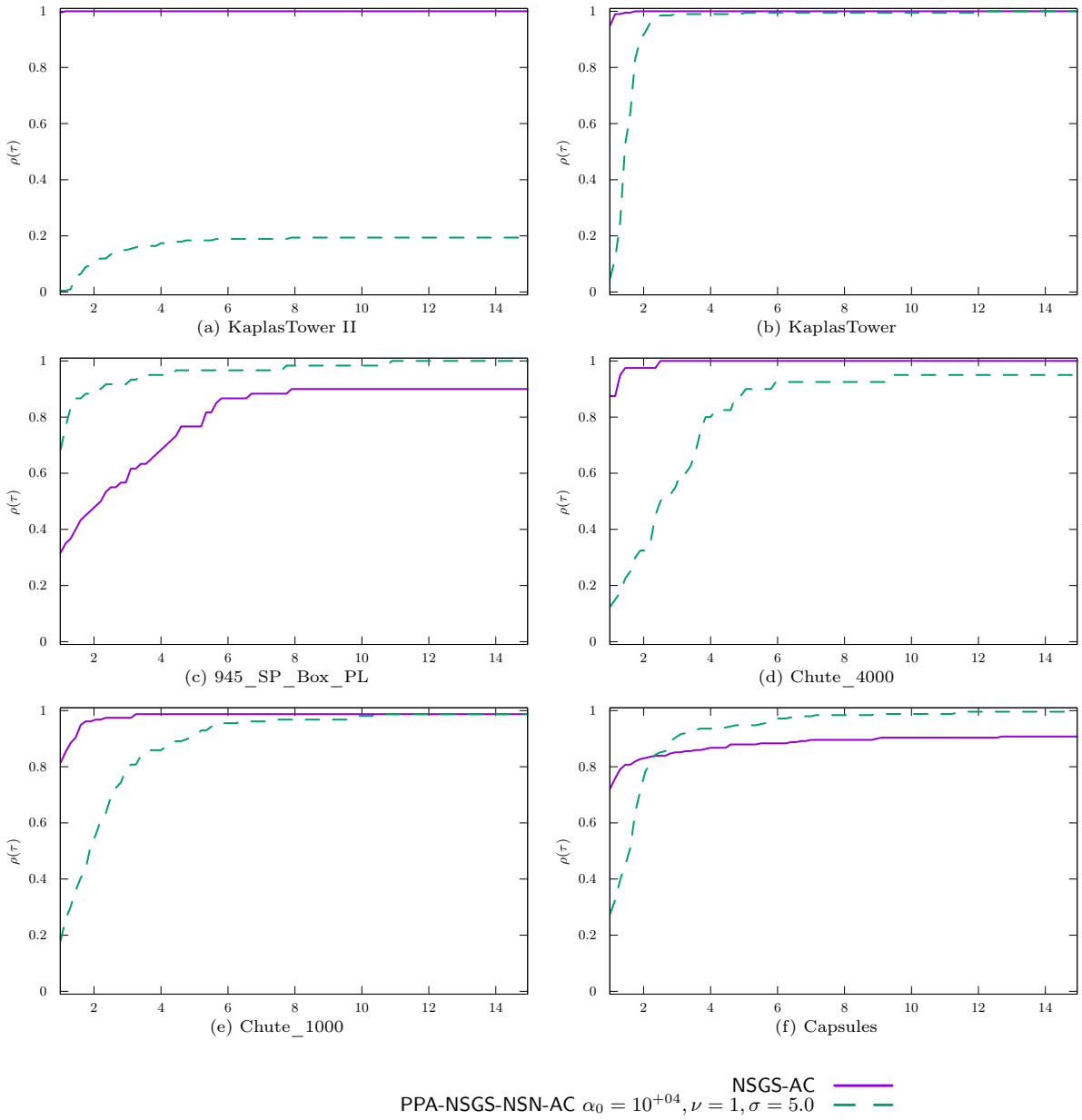


Figure 13: Comparison of internal solvers in PPA-NSGS-$\star$ algorithms.

2. The solvers based on the optimization approach are generally robust but slow. This is mainly due to two reasons. Firstly, we use iterative first order solvers as internal solver with a slow convergence rate. The fact that the Delassus matrix has not full rank in the rigid tests prevents the use of second order methods as nonsmooth Newton methods. For the flexible test, it could be of interest to implement dedicated new solvers of the internal convex problems based on nonsmooth Newton methods. Furthermore, the tests with off-the-shelf implementations of optimization methods were not really concluding. The general convex solvers are not able to exploit the particular structure of the constraints given by a Cartesian product of a large number of second order cones in $\mathbb{R}^3$. Secondly, the fixed point iteration that drives the convergence is generally slow. Once again, it would be valuable to implement a second order method for driving the external loop.

3. On the choice of a specific optimization based strategy with respect to the others, we can observe that the comparison is really problem–dependent. On the test sets Cubes_H8, Bridge_PR, Bridge_PR II, LowWall_FEM and AqueducPR, the ACLM-$\star$ solvers are the best. For the test problems KaplasTower, 945_SP_Box_PL, Chute_4000, Chute_1000 and BoxesStack, the TRESCA-$\star$ solvers are better. Finally, the PANA-$\star$ solvers are better on the 100_PR_PerioBox test set. Since the convex relaxation of the internal problem is made in different manners, it is expected that the different families of solvers behave differently. In particular, if the coefficient of friction is large or if the number of sliding contacts is low, we expect the ACLM-$\star$ solvers to behave better because the s variable in the fixed point iteration will not drastically influence the convergence. On the contrary, when the coefficient of friction is low, we may expect the splitting introduced in the PANA-$\star$ to be better. An analysis of the contact status (closed, sliding, sticking) in the problems would be a next step in understanding the performance of each family.

10 Comparison of different families of solvers.

In this last section, we compare the most efficient solvers for each family. The performance profiles are reported in Figure 15. The main conclusions are as follows:

1. First of all, we can observe that for all the test sets, at least one solver is able to solve all the problems within the prescribed time. Unfortunately, there is no universal solver that outperforms all the other solvers for all the test sets.

2. For the flexible test sets, the nonsmooth Newton solvers NSN-$\star$ are the best solvers. In the test set LowWall_FEM II, the NSN-$\star$ are followed the NSGS-FP-VI-UPK and NSGS-AC solvers. On this test set, the required accuracy is limited to 10^{-04} and the NSGS-$\star$ are still able to reach the tolerance in a competitive time. Between the test sets Cubes_H8 II and Cubes_H8, and between LowWall_FEM II and LowWall_FEM, the required accuracy is decreased to 10^{-08}. With a tighter tolerance, we observe that the relative efficiency of the NSN-$\star$ solvers increase. This was already noted in [Acary et al., 2017]. In other words, on the flexible tests we are able to use nonsmooth Newton methods efficiently since the Delassus matrix W has full rank. In that case, the quadratic convergent rate helps reaching tighter tolerances. Note that in the flexible test sets, the proximal point algorithms PPA-NSN-$\star$ are not really interesting but as the required accuracy decreased, they start to compete with NSGS-$\star$ algorithms.

3. For most of the rigid test sets with a low required accuracy of 10^{-04} as AqueducPR, 945_SP_Box_PL, 100_PR_PerioBox , KaplasTower, Chute_4000 and Chute_1000, the NSGS-$\star$ are the most efficient

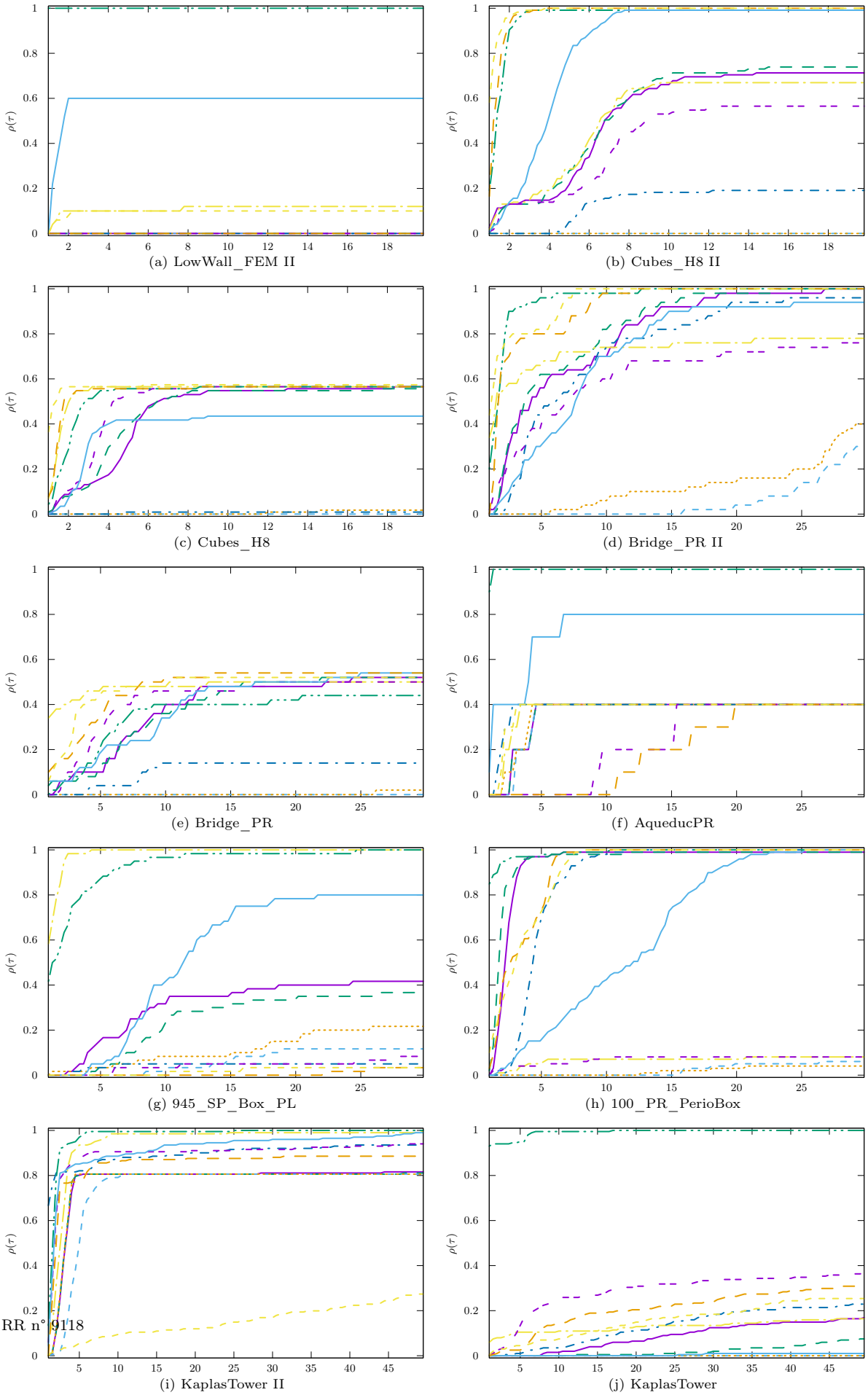

ρ(τ)
(a) LowWall_FEM II
(b) Cubes_H8 II
(c) Cubes_H8
(d) Bridge_PR II
(e) Bridge_PR
(f) AqueducPR
(g) 945_SP_Box_PL
(h) 100_PR_PerioBox
(i) KaplasTower II
(j) KaplasTower

Figure 14

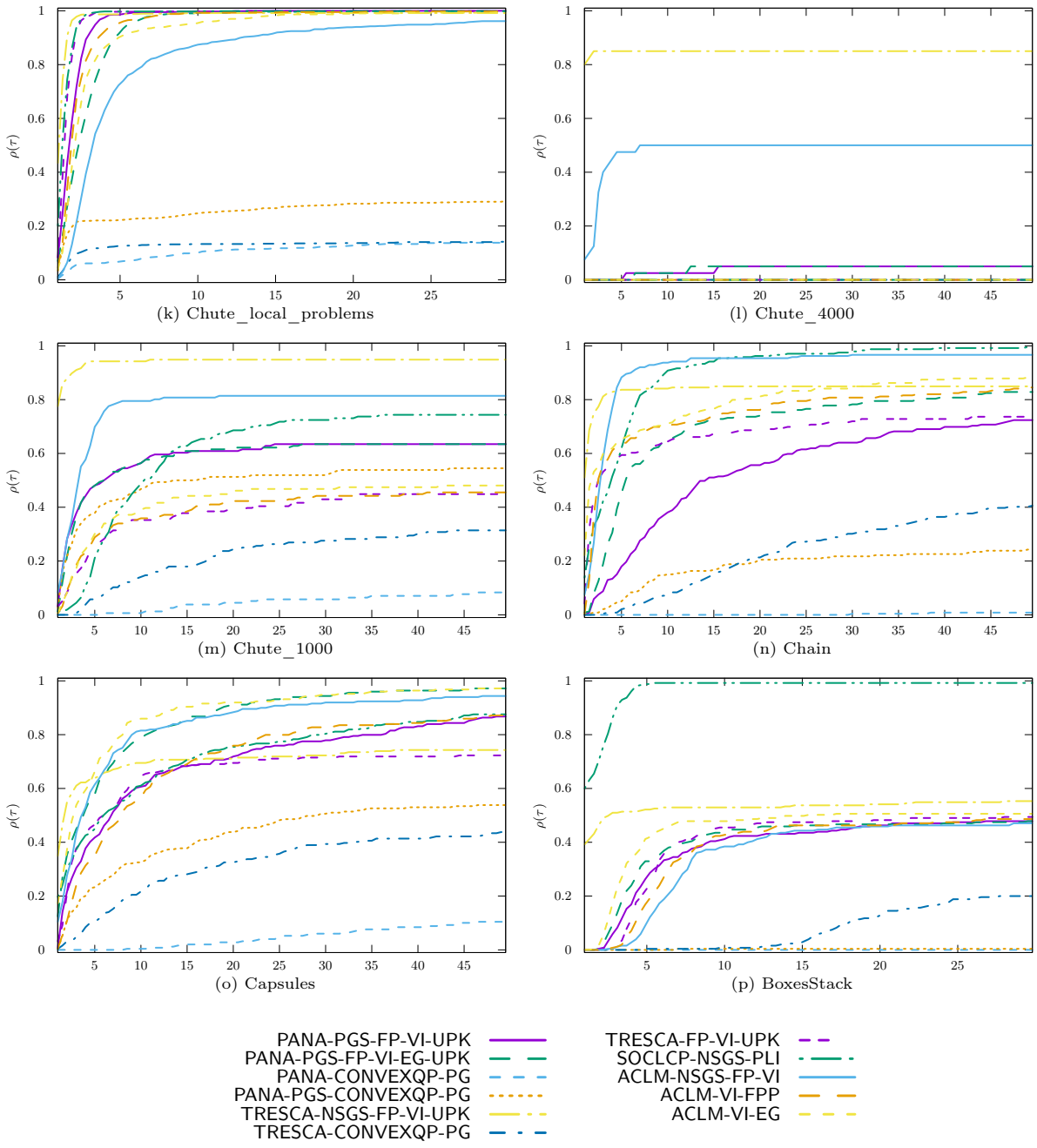


Figure 14: Comparison of the optimization based solvers

and robust solvers. In the case of the test sets Chute_4000 and Chute_1000, the NSGS-FP-VI-UPK solvers are better than the NSGS-AC-⋆ due to some robustness issues in the local solvers based on nonsmooth Newton methods. These solvers are generally followed by optimization based solvers such as ACLM-⋆ and TRESCA-⋆ solvers, except for the test set 945_SP_Box_PL where the more robust solver is TRESCA-NSGS-FP-VI-UPK.

4. For the rigid test sets with a required accuracy of 10^{-08} as Bridge_PR, Chain, Capsules and BoxesStack, the solvers PPA-⋆ are the most efficient and robust solvers. The regularization of the Delassus matrix introduced by the proximal point algorithm has a very positive effect. Especially, it enables the use of nonsmooth Newton techniques that help reaching a tighter accuracy thank to their quadratic convergent rates. The PPA-⋆ algorithms are generally followed by NSGS-⋆, except in the case of the Chain test set where the NSN-⋆ are able to solve 60% of the problems quite efficiently. In the case of the Bridge_PR test set, the use of proximal point technique PPA-NSN-AC-GP $\alpha_0 = 10^{+03}$ is the only one to solve all the problems at the tolerance of 10^{-08}. As discussed in Section 9.4, the rule for updating the proximal point parameter α play an important role and deserves further studies.

5. In the case of the Chute_local_problems test set, we observe that the optimization based solvers are the best and allows one to circumvent the issues of robustness of NSGS-AC-⋆ solvers that are reduced in that case to the NSN-⋆ solvers. We recall that these local problems are extracted from Chute_4000 and selected as most difficult local problems. These problems are characterized by strongly unsymmetric matrices with large extra-diagonal terms compared to the diagonal ones. In that case, the optimization solvers based on a convexification help to solve the problems although the local Delassus matrix is not necessarily symmetric. We can also note as in the Chute_4000 and Chute_1000 that the NSGS-FP-VI-UPK solvers are less sensitive to this asymmetry of the Delassus matrix.

11 General conclusions

In this report, we have reviewed several formulations of the discrete contact problem with Coulomb friction. These formulations open the way to various solving procedures that have been detailed. Some are already well-known: a) the splitting and relaxation techniques (NSGS-⋆ and PSOR-⋆ solvers), b) the nonsmooth Newton methods (NSN-⋆ solvers) and c) the optimization based solvers (PANA-⋆, TRESCA-⋆ and ACLM-⋆ solvers). For the first time, we present general solvers based on the variational inequalities formulation (FP-VI-⋆ and FP-EG-⋆). These methods extend the standard fixed point iteration (FP-DS or also coined Uzawa's algorithm) in various directions and provide some self-adaptive rules to update the ρ parameter that appear to be crucial in practice for the efficiency of the methods. As far as we know, it is also the first application of the proximal point algorithms (PPA-⋆) to the discrete frictional contact problem. This new family of solvers appears to be a promising alternative when we want to reach tight accuracy for collections of rigid bodies such as granular materials.

Then we presented a thorough comparison of solvers over a large set of test problems. Using performance profiles, the solvers have been compared family by family, and then altogether. The main conclusions and perspectives of this study are as follows:

- The methods based on variational inequality formulations (FP-VI-⋆) are robust, if a consistent self-adaptive rule for the parameter ρ is used. We presented two rules that yield very satisfactory results. Thanks to their robustness, these methods provide reliable solvers for the local problem

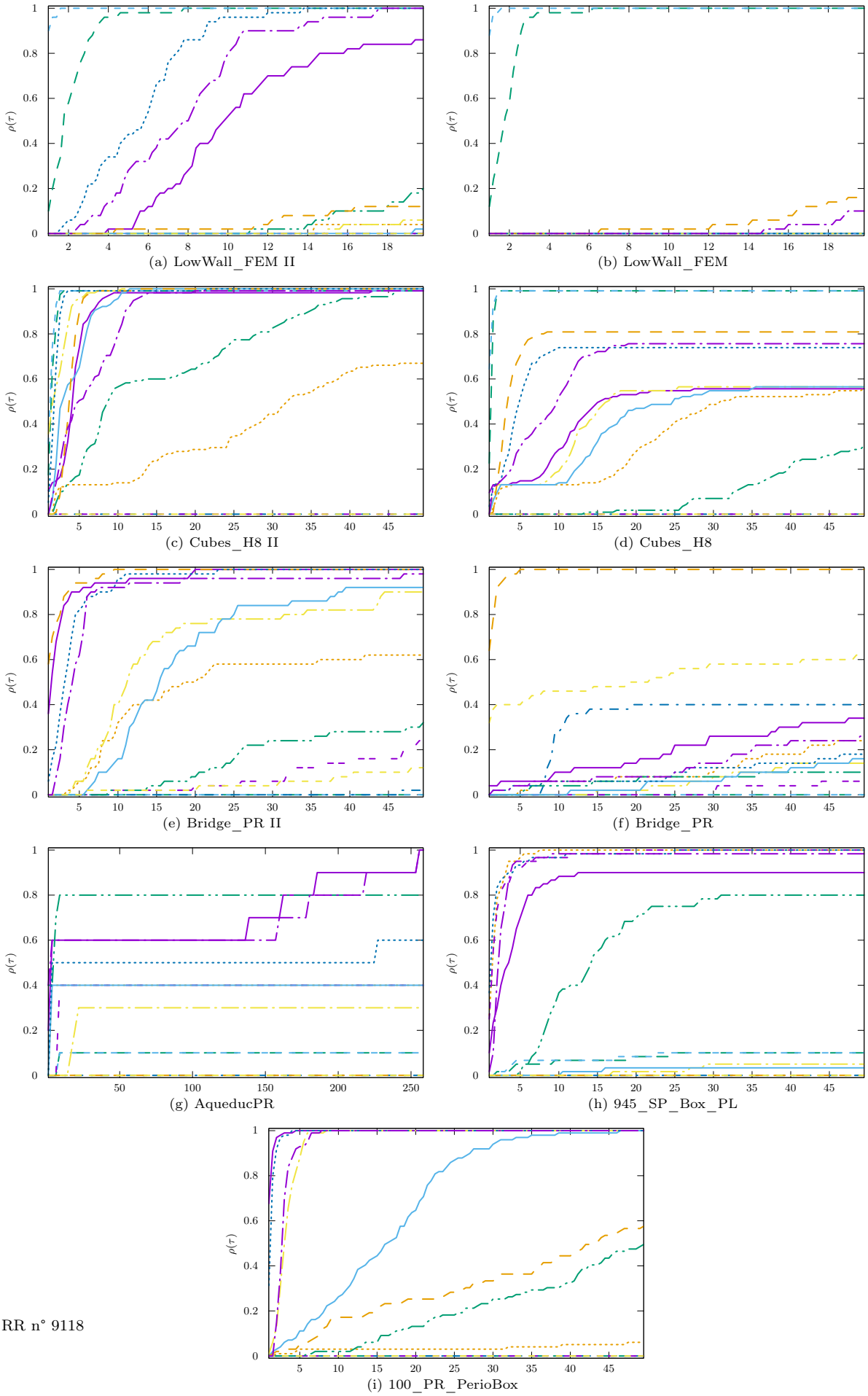


Figure 15: Comparison of the solvers between families

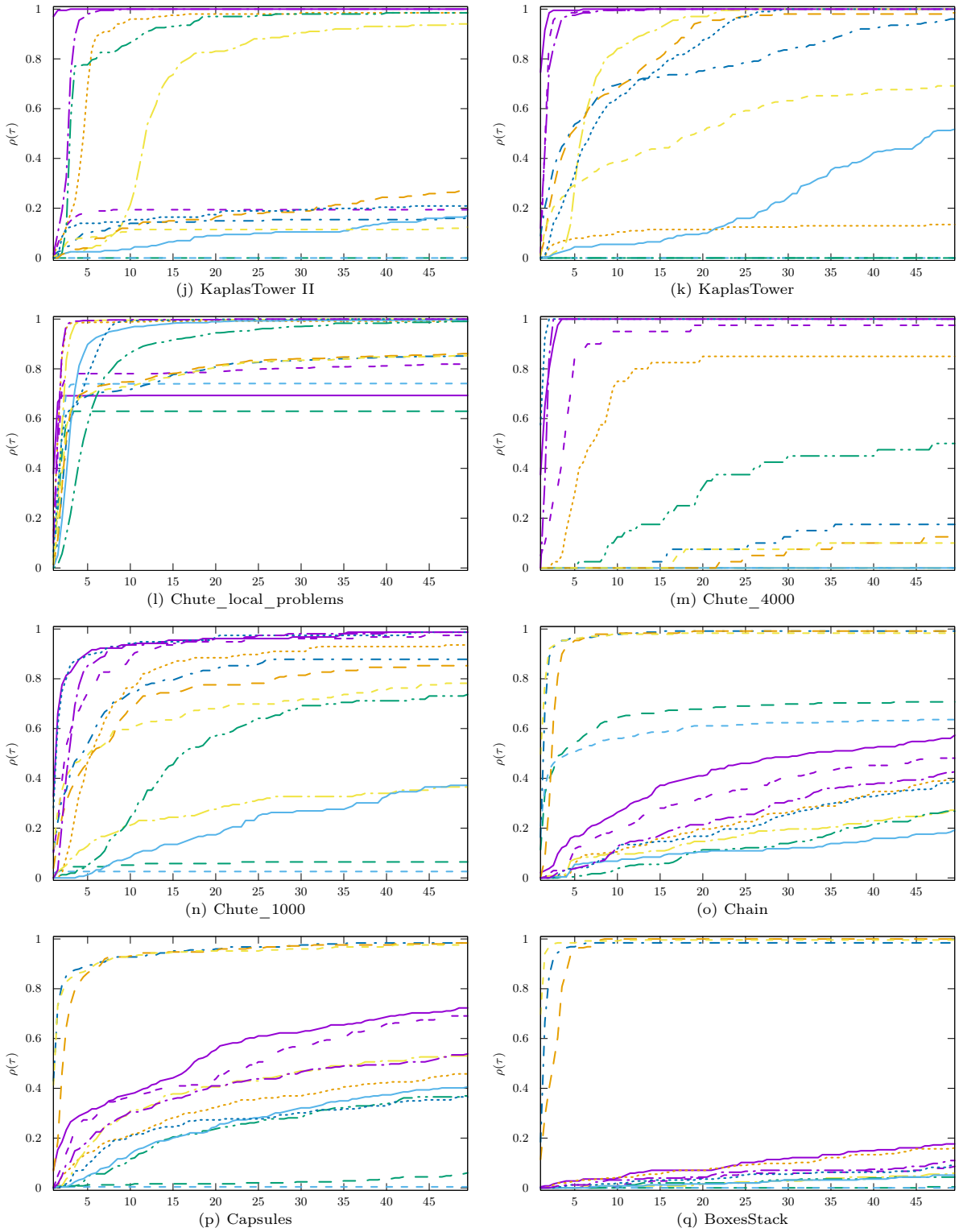


Figure 15: Comparison of the solvers between families (continued)

in splitting techniques. Nevertheless, the convergence is slow: those methods have difficulties to get a solution within the prescribed time for tight tolerances, or if the problem size is large. The main perspectives for these methods are a) to adapt the values of ρ contact by contact to try to improve the convergence speed and b) to perform computation in parallel for large scale systems. Indeed, each iteration of the FP-VI-$\star$ solvers may be straightforwardly implemented on distributed computer architectures.

- The methods based on splitting techniques, the NSGS-$\star$ solvers, provide us with robust and efficient solvers provided that the local solver is robust. They are generally more efficient than the FP-VI-$\star$ methods since they exploit the particular structure of the problem (sparse block sparsity and local solver routines). However, they suffer from the same problems as the FP-VI-$\star$ solvers: the convergence rate is low and high accuracy is difficult to reach within the prescribed time. The main perspective for this solver is to improve the robustness and the efficiency of the local solver, for instance by using proximal point techniques or optimization based solvers. Regarding the PSOR-$\star$ solvers, for some values of the relaxation parameter ω, the convergence rate is greatly improved with respect to the NSGS-$\star$ solvers. However, guessing the correct value of the parameter ω is challenging as some values may increase the computational effort or make the algorithm diverge. Clearly, a self adaptive rule for sizing the relaxation parameter ω would be a notable improvement.

- The nonsmooth Newton solvers NSN-$\star$ appear to be a very efficient family of solvers for problems that have a full-rank Delassus matrix or a very low contact density. For instance, in the case of the flexible tests, they are the best solvers among others and they are able to reach tight tolerances that are not reachable with the FP-VI-$\star$ and NSGS-$\star$ solvers. For the other test sets, they suffer from robustness issues. To overcome this, we work on several options: a) the choice of the ρ parameters in the equation based formulation, b) the line-search procedures may help to stabilize the convergence at the price to slow down the convergence and c) improving the initial starting point of the solver with a FP-VI-$\star$. All these improvements appear to increase the robustness. Unfortunately, it was not sufficient to circumvent all the divergence problems. Some pointers in the literature try to modify the iteration matrix in the Newton loop to improve robustness when the iterates are far form the solution. This solution has not been tested. The main perspectives for these solvers are to improve their robustness by testing modifications of the iteration matrix or self-adapting rule for sizing ρ. The question of the scaling and the preconditioning must be studied deeper. When the solvers are robust, these solvers are also highly parallelizable for large systems since we can rely on massively parallel solvers for linear systems such as MUMPS.

- As we discussed before, the PPA-$\star$ solvers are a possible solution for improving the robustness of the NSN-$\star$ methods while keeping their convergence rates. This solution proves its efficiency on a lot of test sets. Nevertheless, we were not able to find an universal rule for updating the parameter α such that it works for all test sets. Clearly, this deserves more studies on this aspect.

- The optimization based solvers (PANA-$\star$, TRESCA-$\star$ and ACLM-$\star$ might also exhibit good robustness properties. Unfortunately, they suffer from the slow convergence of the external loop based on a fixed point updating, which is not compensated by the efficiency of the convex problem solver. As we have seen, the nonconvexity of the problems is not the only difficulty: most of the time the rank deficiency of the Delassus matrix is the main cause of the slow convergence or divergence. Finally, it would be worthwhile investigating why an optimization formulation is better than another for some test sets. One of the reasons might be that the contact status (closed, sticking, sliding) are not distributed in the same way along the test sets. A study based on the contact status would be

complementary to the measure of the rank ratio and the contact density for guessing the cause of the issues.

References

V. Acary and B. Brogliato. *Numerical methods for nonsmooth dynamical systems. Applications in mechanics and electronics.* Lecture Notes in Applied and Computational Mechanics 35. Berlin: Springer. xxi, 525 p. , 2008. 7, 12

V. Acary and F. Cadoux. *Recent Advances in Contact Mechanics, Stavroulakis, Georgios E. (Ed.)*, volume 56 of *Lecture Notes in Applied and Computational Mechanics*, chapter Applications of an existence result for the Coulomb friction problem. Springer Verlag, 2013. 7, 14, 23

V. Acary, F. Cadoux, C. Lemaréchal, and J. Malick. A formulation of the linear discrete coulomb friction problem via convex optimization. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 91(2):155–175, 2011. ISSN 1521-4001. doi: 10.1002/zamm.201000073. URL `http://dx.doi.org/10.1002/zamm.201000073`. 7, 8, 12, 14, 23

V. Acary, M. Brémond, T. Koziara, and F. Pérignon. FCLIB: a collection of discrete 3D Frictional Contact problems. Technical Report RT-0444, INRIA, February 2014. URL `https://hal.inria.fr/hal-00945820`. 40

V. Acary, M. Brémond, O. Huber, and F. Pérignon. An introduction to Siconos. Technical Report TR-0340, second version, INRIA, http://hal.inria.fr/inria-00162911/en/, 2015. 9

Vincent Acary, Maurice Brémond, and Frédéric Dubois. Méthodes de Newton non–lisses pour les problèmes de contact frottant dans les systèmes de multi–corps flexibles. In *CSMA 2017 - 13ème Colloque National en Calcul des Structures*, page 8, Presqu'île de Giens (Var), France, May 2017. URL `https://hal.inria.fr/hal-01562706`. 61

A.M. Al-Fahed, G.E. Stavroulakis, and P.D. Panagiotopulos. Hard and soft fingered robot grippers. the linear complementarity approach. *Zeitschrift für Angewandte Mathematik und Mechanik*, 71:257–265, 1991. 8

P. Alart. Injectivity and surjectivity criteria for certain mappings of $\mathbb{R}^n$ into itself; application to contact mechanics. (Critères d'injectivité et de surjectivité pour certaines applications de $\mathbb{R}^n$ dans lui-même; application à la mécanique du contact.). *RAIRO, Modélisation Math. Anal. Numér.*, 27(2):203–222, 1993. 28

P. Alart. Méthode de newton généralisée en mécanique du contact. *Journal de Mathématiques Pures et Appliquées*, 1995. 28

P. Alart and A. Curnier. A mixed formulation for frictional contact problems prone to Newton like solution method. *Computer Methods in Applied Mechanics and Engineering*, 92(3):353–375, 1991. 18

M. Anitescu and F.A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics, Transactions of A.S.M.E.*, 14:231–247, 1997. 7, 8

M. Anitescu and A. Tasora. An iterative approach for cone complementarity problems for nonsmooth dynamics. *Comput. Optim. Appl.*, 47(2):207–235, 2010. ISSN 0926-6003. doi: 10.1007/s10589-008-9223-4. URL `http://dx.doi.org/10.1007/s10589-008-9223-4`. 8, 38

H.J.C. Barbosa and R.A. Feijóo. A numerical algorithm for signorini problem with Coulomb friction. In Del Piero and Maceri [1985]. 22

H.J.C. Barbosa, F.M.P. Raupp, and C.C.H. Borges. Numerical experiments with algorithms for bound constrained quadratic programming in Mechanics. *Computers & Structures*, 64(1–4):579–594, 1997. 36

J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, and C.A. Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer-Verlag, 2003. 29

Olivier Bonnefon and Gilles Daviet. Quartic formulation of Coulomb 3D frictional contact. Technical Report RT-0400, INRIA, January 2011. URL `https://hal.inria.fr/inria-00553859`. 32

P. Breitkopf and M. Jean. Modélisation parallèèle des matéériaux granulaires. In *Actes du 4ème Colloque National en Calcul des Structures*, pages 387–392, Giens(Var), 18-21 Mai 1999. 8

F. Cadoux. *Analyse convexe et optimisation pour la dynamique non–regulière*. PhD thesis, Université Joseph Fourier, Grenoble I, 2009. 7, 23

P.H. Calamai and J.J. More. Projected gradient methods for linearly constrained problems. *Mathematical Programming*, 39(1):93–116, 1987. ISSN 0025-5610. doi: http://dx.doi.org/10.1007/BF02592073. 35

P. Chabrand, F. Dubois, and M. Raous. Various numerical methods for solving unilateral contact problems with friction. *Mathematical and Computer Modelling*, 28(4–8):97 – 108, 1998. ISSN 0895-7177. doi: http://dx.doi.org/10.1016/S0895-7177(98)00111-3. URL `http://www.sciencedirect.com/science/article/pii/S0895717798001113`. Recent Advances in Contact Mechanics. 9

A. Chaudhary and K.J. Bathe. A solution method for static and dynamic analysis of three–dimensional contact problems with friction. *Computers & Structures*, 24(6):855–873, 1986. 6

P. Christensen, A. Klarbring, J. Pang, and N. Stromberg. Formulation and comparison of algorithms for frictional contact problems. *International Journal for Numerical Methods in Engineering*, 42:145–172, 1998. 9, 18

P.W. Christensen and J.S. Pang. Frictional contact algorithms based on semismooth newton methods. In M. Fukushima & L. Qi, editor, *Reformulation - Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, pages 81–116, Dordrecht, 1998. Kluwer Academic Publishers. 8, 27

A. Curnier and P. Alart. A generalized Newton method for contact problems with friction. *Journal de Mécanique Théorique et Appliquée*, supplément no 1 to 7:67–82, 1988. 18

Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. A Hybrid Iterative Solver for Robustly Capturing Coulomb Friction in Hair Dynamics. *ACM Transactions on Graphics*, 30(6):139:1–139:12, 2011. doi: 10.1145/2070781.2024173. URL `https://hal.inria.fr/hal-00647497`. 47

G. De Saxcé. Une généralisation de l'inégalité de Fenchel et ses applications aux lois constitutives. *Comptes Rendus de l'Académie des Sciences*, t 314,série II:125–129, 1992. 12, 21

G. De Saxcé and Z.-Q. Feng. New inequality and functional for contact with friction : The implicit standard material approach. *Mech. Struct. & Mach.*, 19(3):301–325, 1991. 12, 23

G. De Saxcé and Z.-Q. Feng. The bipotential method: A constructive approach to design the complete contact law with friction and improved numerical algorithms. *Mathemetical and Computer Modelling*, 28(4):225–245, 1998. 23

G. Del Piero and F. Maceri, editors. *Unilateral Problems in Structural Analysis*, volume 288 of *CISM courses and lectures*, Ravello, Italy, September 22–24 1983. Springer. 6

G. Del Piero and F. Maceri, editors. *Unilateral Problems in Structural Analysis – II.*, volume 304 of *CISM courses and lectures*, Prescudin, Italy, June 17–20 1985. Springer. 6, 69

E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002. 39

Z. Dostál. Box constrained quadratic programming with proportioning and projections. *SIAM J. Optim.*, 7(3):871–887, 1997. ISSN 1052-6234. doi: 10.1137/S1052623494266250. URL `http://dx.doi.org/10.1137/S1052623494266250`. 37

Z. Dostál and T. Kozubek. An optimal algorithm and superrelaxation for minimization of a quadratic function subject to separable convex constraints with applications. *Math. Program.*, 135(1-2, Ser. A): 195–220, 2012. ISSN 0025-5610. doi: 10.1007/s10107-011-0454-2. URL `http://dx.doi.org/10.1007/s10107-011-0454-2`. 36, 38

Z. Dostál and R. Kučera. An optimal algorithm for minimization of quadratic functions with bounded spectrum subject to separable convex inequality and linear equality constraints. *SIAM Journal on Optimization*, 20(6):2913–2938, 2010. doi: 10.1137/090751414. URL `http://dx.doi.org/10.1137/090751414`. 38

Z. Dostál, J. Haslinger, and R. Kučera. Implementation of the fixed point method in contact problems with Coulomb friction based on a dual splitting type technique. *J. Comput. Appl. Math.*, 140(1-2): 245–256, 2002. ISSN 0377-0427. doi: 10.1016/S0377-0427(01)00405-8. URL `http://dx.doi.org/10.1016/S0377-0427(01)00405-8`. 37

Zdeněk Dostál. *Scalable Algorithms for Contact Problems*. Springer New York, New York, NY, 2016. ISBN 978-1-4939-6834-3. doi: 10.1007/978-1-4939-6834-3_8. URL `https://doi.org/10.1007/978-1-4939-6834-3_8`. 35, 38

Zdeněk Dostál and Joachim Schöberl. Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination. *Comput. Optim. Appl.*, 30(1):23–43, 2005. ISSN 0926-6003. doi: 10.1007/s10589-005-4557-7. URL `http://dx.doi.org/10.1007/s10589-005-4557-7`. 38

Z. Dostál, T. Kozubek, P. Horyl, T. Brzobohatý, and A. Markopoulos. A scalable tfeti algorithm for two-dimensional multibody contact problems with friction. *Journal of Computational and Applied Mathematics*, 235(2):403 – 418, 2010. ISSN 0377-0427. doi: https://doi.org/10.1016/j.cam.2010.05.042. URL `http://www.sciencedirect.com/science/article/pii/S0377042710003328`. Special Issue on Advanced Computational Algorithms. 8

F. Facchinei and J. S. Pang. *Finite-dimensional Variational Inequalities and Complementarity Problems*, volume I & II of *Springer Series in Operations Research*. Springer Verlag NY. Inc., 2003. 6, 15, 16, 17, 27

R. Fletcher. *Practical Methods of Optimization*. Chichester: John Wiley & Sons, Inc., 1987. 35

David Chin-Lung Fong and Michael Saunders. Lsmr: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, 33(5):2950–2971, 2011. doi: 10.1137/10079687X. URL `https://doi.org/10.1137/10079687X`. 41

M. Fortin and R. Glowinski. *Augmented Lagrangian methods*, volume 15 of *Studies in Mathematics and its Applications*. North-Holland Publishing Co., Amsterdam, 1983. ISBN 0-444-86680-9. Applications to the numerical solution of boundary value problems, Translated from the French by B. Hunt and D. C. Spicer. 24

M. Fukushima, Z.Q. Luo, and P. Tseng. Smoothing functions for second-order-cone complementarity problems. *SIAM Journal on Optimization*, 12(2):436–460, 2001. doi: 10.1137/S1052623400380365. URL `http://link.aip.org/link/?SJE/12/436/1`. 19, 20

R. Glowinski, Lions. J.L., and R. Trémoliéres. *Approximations des Inéquations Variationnelles*. Dunod, Paris, 1976. 24

William W. Hager and Hongchao Zhang. Self-adaptive inexact proximal point methods. *Computational Optimization and Applications*, 39(2):161–181, Mar 2008. ISSN 1573-2894. doi: 10.1007/s10589-007-9067-3. URL `https://doi.org/10.1007/s10589-007-9067-3`. 34

Deren Han and Hong K. Lo. Two new self-adaptive projection methods for variational inequality problems. *Computers & Mathematics with Applications*, 43(12):1529 – 1537, 2002. ISSN 0898-1221. doi: http://dx.doi.org/10.1016/S0898-1221(02)00116-5. URL `http://www.sciencedirect.com/science/article/pii/S0898122102001165`. 25

P.T. Harker and J.-S. Pang. Finite-dimensional variational inequality and complementarity problems: a survey of theory, algorithms and applications. *Mathematical Programming*, 48:160–220, 1990. 15

J. Haslinger. Approximation of the signorini problem with friction, obeying the coulomb law. *Mathematical Methods in the Applied Sciences*, 5:422–437, 1983. 6, 23, 36

J. Haslinger. Least square method for solving contact problems with friction obeying coulomb's law. *Applications of mathematics*, 29(3):212–224, 1984. URL `http://dml.cz/dmlcz/104086`. 6, 23, 36

J. Haslinger and P. D. Panagiotopoulos. The reciprocal variational approach to the signorini problem with friction. approximation results. *Proceedings of the Royal Society of Edinburgh: Section A Mathematics*, 98:365–383, 1984. ISSN 1473-7124. doi: 10.1017/S0308210500013536. URL `http://journals.cambridge.org/article_S0308210500013536`. 6, 22

J. Haslinger, I. Hlaváček, and J. Nečas. Numerical methods for unilateral problems in solid mechanics. In P.G. Ciarlet and J.L. Lions, editors, *Handbook of Numerical Analysis*, volume IV, Part 2, pages 313–485, Amsterdam, 1996, 1996. North-Holland. 22, 23, 36

J. Haslinger, Z. Dostál, and R. Kučera. On a splitting type algorithm for the numerical realization of contact problems with Coulomb friction. *Comput. Methods Appl. Mech. Engrg.*, 191(21-22):2261–2281, 2002. ISSN 0045-7825. doi: 10.1016/S0045-7825(01)00378-4. URL `http://dx.doi.org/10.1016/S0045-7825(01)00378-4`. 37

J. Haslinger, R. Kučera, and Zdeněk D. An algorithm for the numerical realization of 3D contact problems with Coulomb friction. In *Proceedings of the 10th International Congress on Computational and Applied Mathematics (ICCAM-2002)*, volume 164/165, pages 387–408, 2004. doi: 10.1016/j.cam.2003.06.002. URL `http://dx.doi.org/10.1016/j.cam.2003.06.002`. 8, 37

J. Haslinger, R. Kučera, O. Vlach, and C.C. Baniotopoulos. Approximation and numerical realization of 3d quasistatic contact problems with coulomb friction. *Mathematics and Computers in Simulation*, 82 (10):1936 – 1951, 2012. ISSN 0378-4754. doi: http://dx.doi.org/10.1016/j.matcom.2011.01.004. URL `http://www.sciencedirect.com/science/article/pii/S0378475411000310`. The Fourth IMACS Conference : Mathematical Modelling and Computational Methods in Applied Sciences and Engineering" Devoted to Owe Axelsson in ocassion of his 75th birthday. 37

S. Hayashi, N. Yamashita, and M. Fukushima. A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM J. on Optimization*, 15(2):593–615, 2005. ISSN 1052-6234. doi: http://dx.doi.org/10.1137/S1052623403421516. 20, 78

B.S. He and L.Z. Liao. Improvements of some projection methods for monotone nonlinear variational inequalities. *Journal of Optimization Theory and Applications*, 112(1):111–128, 2002. ISSN 0022-3239. doi: 10.1023/A:1013096613105. URL `http://dx.doi.org/10.1023/A%3A1013096613105`. 25

M.R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4: 303–320, 1969. 18

T. Heyn, M. Anitescu, A. Tasora, and D. Negrut. Using Krylov subspace and spectral methods for solving complementarity problems in many-body contact dynamics simulation. *Internat. J. Numer. Methods Engrg.*, 95(7):541–561, 2013. ISSN 0029-5981. doi: 10.1002/nme.4513. URL `http://dx.doi.org/10.1002/nme.4513`. 8

T.D. Heyn. *On the modeling, simulation, and visualization of many–body dynamics problems with friction and contact*. PhD thesis, University of Wisconsin–Madison, 2013. 8

S. Hüeber, G. Stadler, and B. I. Wohlmuth. A primal-dual active set algorithm for three-dimensional contact problems with coulomb friction. *SIAM J. Sci. Comput.*, 30(2):572–596, February 2008. ISSN 1064-8275. doi: 10.1137/060671061. URL `http://dx.doi.org/10.1137/060671061`. 19, 27, 28

M. Jean and J.J. Moreau. Dynamics in the presence of unilateral contacts and dry friction: a numerical approach. In G. Del Pietro and F. Maceri, editors, *Unilateral problems in structural analysis. II*, pages 151–196. CISM 304, Spinger Verlag, 1987. 6, 18

P. Joli and Z.-Q. Feng. Uzawa and newton algorithms to solve frictional contact problems within the bi-potential framework. *International Journal for Numerical Methods in Engineering*, 73(3):317–330, 2008. ISSN 1097-0207. doi: 10.1002/nme.2073. URL `http://dx.doi.org/10.1002/nme.2073`. 27

F. Jourdan, P. Alart, and M. Jean. A Gauss Seidel like algorithm to solve frictional contact problems. *Computer Methods in Applied Mechanics and Engineering*, 155(1):31–47, 1998. 28, 32

Michael G. Katona. A simple contact–friction interface element with applications to buried culverts. *International Journal for Numerical and Analytical Methods in Geomechanics*, 7(3):371–384, 1983. ISSN 1096-9853. doi: 10.1002/nag.1610070308. URL `http://dx.doi.org/10.1002/nag.1610070308`. 6

E.N. Khobotov. Modification of the extra-gradient method for solving variational inequalities and certain optimization problems. *{USSR} Computational Mathematics and Mathematical Physics*, 27(5):120 – 127, 1987. ISSN 0041-5553. doi: http://dx.doi.org/10.1016/0041-5553(87)90058-9. URL `http://www.sciencedirect.com/science/article/pii/0041555387900589`. 25

N. Kikuchi and J. T. Oden. *Contact problems in elasticity: a study of variational inequalities and finite element methods*, volume 8 of *SIAM Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1988. ISBN 0-89871-202-5. doi: 10.1137/1.9781611970845. URL `http://dx.doi.org/10.1137/1.9781611970845`. 8

A. Klarbring. A mathematical programming approach to three-dimensional contact problem with friction. *Compt. Methods Appl. Math. Engrg.*, 58:175–200, 1986. 8

Anders Klarbring and Jong-Shi Pang. Existence of solutions to discrete semicoercive frictional contact problems. *SIAM Journal on Optimization*, 8(2):414–442, 1998. ISSN 1052-6234. 7, 14

Jan Kleinert, Bernd Simeon, and Martin Obermayr. An inexact interior point method for the large-scale simulation of granular material. *Computer Methods in Applied Mechanics and Engineering*, 278 (0):567 – 598, 2014. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/j.cma.2014.06.009. URL `http://www.sciencedirect.com/science/article/pii/S0045782514001959`. 8

G.M. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12 (747–756), 1976. 24, 25

T. Koziara and N. Bićanić. A distributed memory parallel multibody contact dynamics code. *International Journal for Numerical Methods in Engineering*, 87(1-5):437–456, 2011. ISSN 1097-0207. doi: 10.1002/nme.3158. URL `http://dx.doi.org/10.1002/nme.3158`. 8

Tomasz Koziara and Nenad Bićanić. Semismooth newton method for frictional contact between pseudo-rigid bodies. *Computer Methods in Applied Mechanics and Engineering*, 197(33–40):2763 – 2777, 2008. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/j.cma.2008.01.006. URL `http://www.sciencedirect.com/science/article/pii/S0045782508000194`. 19, 28, 29

K. Krabbenhoft, A.V. Lyamin, J. Huang, and M. Vicente da Silva. Granular contact dynamics using mathematical programming methods. *Computers and Geotechnics*, 43:165 – 176, 2012. ISSN 0266-352X. doi: http://dx.doi.org/10.1016/j.compgeo.2012.02.006. URL `http://www.sciencedirect.com/science/article/pii/S0266352X12000262`. 8

R. Kučera. Minimizing quadratic functions with separable quadratic constraints. *Optim. Methods Softw.*, 22(3):453–467, 2007. ISSN 1055-6788. doi: 10.1080/10556780600609246. URL `http://dx.doi.org/10.1080/10556780600609246`. 37

Radek Kučera. Convergence rate of an optimization algorithm for minimizing quadratic functions with separable convex constraints. *SIAM J. Optim.*, 19(2):846–862, 2008. ISSN 1052-6234. doi: 10.1137/060670456. URL `http://dx.doi.org/10.1137/060670456`. 37

T.A. Laursen. *Computational Contact and Impact Mechanics – Fundamentals of Modeling Interfacial Phenomena in Nonlinear Finite Element Analysis*. Springer Verlag, 2003. 1st ed. 2002. Corr. 2nd printing,. 7, 12

A.Y.T. Leung, C. Guoqing, and C. Wanji. Smoothing Newton method for solving two– and thre–dimensional frictional contact problems. *International Journal for Numerical Methods in Engineering*, 41:1001–1027, 1998. 19

AnandR. Mijar and JasbirS. Arora. Study of variational inequality and equality formulations for elasto-static frictional contact problems. *Archives of Computational Methods in Engineering*, 7(4):387–449,

2000a. ISSN 1134-3060. doi: 10.1007/BF02736213. URL http://dx.doi.org/10.1007/BF02736213. 9

A.R. Mijar and J.S. Arora. Review of formulations for elastostatic frictional contact problems. *Structural and Multidisciplinary Optimization*, 20(3):167–189, 2000b. ISSN 1615-147X. doi: 10.1007/s001580050147. URL http://dx.doi.org/10.1007/s001580050147. 9

A.R. Mijar and J.S. Arora. An augmented lagrangian optimization method for contact analysis problems, 1: formulation and algorithm. *Structural and Multidisciplinary Optimization*, 28(2-3):99–112, 2004a. ISSN 1615-147X. doi: 10.1007/s00158-004-0423-y. URL http://dx.doi.org/10.1007/s00158-004-0423-y. 9

A.R. Mijar and J.S. Arora. An augmented lagrangian optimization method for contact analysis problems, 2: numerical evaluation. *Structural and Multidisciplinary Optimization*, 28(2-3):113–126, 2004b. ISSN 1615-147X. doi: 10.1007/s00158-004-0424-x. URL http://dx.doi.org/10.1007/s00158-004-0424-x. 9

E.N. Mitsopoulou and I.N. Doudoumis. A contribution to the analysis of unilateral contact problems with friction. *Solid Mechanics Archives*, 12(3):165–186, 1987. 32

E.N. Mitsopoulou and I.N. Doudoumis. On the solution of the unilateral contact frictional problem for general static loading conditions. *Computers & Structures*, 30(5):1111–1126, 1988. 6, 32

Tomoshi Miyamura, Yoshihiro Kanno, and Makoto Ohsaki. Combined interior-point method and semismooth newton method for frictionless contact problems. *International Journal for Numerical Methods in Engineering*, 81(6):701–727, 2010. ISSN 1097-0207. doi: 10.1002/nme.2707. URL http://dx.doi.org/10.1002/nme.2707. 8

José Luis Morales, Jorge Nocedal, and Mikhail Smelyanskiy. An algorithm for the fast solution of symmetric linear complementarity problems. *Numerische Mathematik*, 111(2):251–266, 2008. ISSN 0945-3245. doi: 10.1007/s00211-008-0183-5. URL http://dx.doi.org/10.1007/s00211-008-0183-5. 8

J.J. Moré and G. Toraldo. On the solution of large quadratic convex programming problems with bound constraints. *SIAM J. Optimization*, 1(1):93–113, 1991. 35, 37

Jorge J. Moré and Gerardo Toraldo. Algorithms for bound constrained quadratic programming problems. *Numerische Mathematik*, 55(4):377–400, Jul 1989. ISSN 0945-3245. doi: 10.1007/BF01396045. URL https://doi.org/10.1007/BF01396045. 35

J.J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la société mathématique de France*, 93:273–299, 1965. 32

J.J. Moreau. Unilateral contact and dry friction in finite freedom dynamics. In J.J. Moreau and Panagiotopoulos P.D., editors, *Nonsmooth Mechanics and Applications*, number 302 in CISM, Courses and lectures, pages 1–82. CISM 302, Spinger Verlag, Wien- New York, 1988. Formulation mathematiques tire du livre Contacts mechanics. 11

H. Mylapilli and A. Jain. Complementarity techniques for minimal coordinates contact dynamics. *ASME Journal of Computational and Nonlinear Dynamics*, 12(2), 2017. 9

J. Nečas, J. Jarušek, and J. Haslinger. On the solution of the variational inequality to the Signorini problem with small friction. *Bollettino U.M.I.*, 5(17-B):796–811, 1980. 6, 23, 36

J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Verlag, 1999. 35

P.D. Panagiotopoulos. A nonlinear programming approach to the unilateral contact-, and friction-boundary value problem in the theory of elasticity. *Ingenieur-Archiv*, 44(6):421–432, 1975. ISSN 0020-1154. doi: 10.1007/BF00534623. URL `http://dx.doi.org/10.1007/BF00534623`. 6, 22

J.S. Pang and J.C. Trinkle. Complementarity formulations and existence of solutions of dynamic multi-rigid-body contact problems with Coulomb friction. *Mathematical Programming*, 73:199–226, 1996. 7, 8

Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1 (3):127–239, 2014. 32

J.K. Park and B.M. Kwak. Three dimensional frictional contact analysis using the homotopy method. *Journal of Applied Mechanics, Transactions of A.S.M.E*, 61:703–709, 1994. 19

L. Qi and J. Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58:353–367, 1993. 27

L. Qi, D. Sun, and M. Ulbrich, editors. *Semismooth and Smoothing Newton Methods*. Springer Verlag, 2018. 27

M. Raous, P. Chabrand, and F. Lebon. Numerical methods for frictional contact problems and applications. *J. Méc. Théor. Appl.*, 7(1):111–18, 1988. 9

M. Renouf, F. Dubois, and P. Alart. A parallel version of the Non Smooth Contact Dynamics algorithm applied to the simulation of granular media. *J. Comput. Appl. Math.*, 168:375–38, 2004. 8

R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976. 33

R.T. Rockafellar. Augmented Lagrange multiplier functions and duality in nonconvex programming. *SIAM Journal on Control*, 12:268–285, 1974. 18

R.T. Rockafellar. Lagrange multipliers and optimality. *SIAM Review*, 35(2):183–238, 1993. ISSN 0036-1445. 18

R.T. Rockafellar and R.J.B. Wets. *Variational Analysis*, volume 317. Springer Verlag, New York, 1997. 77

G. De Saxcé and Z.-Q. Feng. The bipotential method: A constructive approach to design the complete contact law with friction and improved numerical algorithms. *Mathematical and Computer Modelling*, 28(4–8):225 – 245, 1998. ISSN 0895-7177. doi: http://dx.doi.org/10.1016/S0895-7177(98)00119-8. URL `http://www.sciencedirect.com/science/article/pii/S0895717798001198`. Recent Advances in Contact Mechanics. 12

M. Sibony. Méthodes itératives pour les équations et inéquations aux dérivées partielles non linéaires de type monotone. *Calcolo*, 7:65—183, 1970. 17

J.C. Simo and T.A. Laursen. An augmented Lagrangian treatment of contact problems involving friction. *Computers & Structures*, 42(1):97–116, 1992. 18, 24

M.V. Solodov and P. Tseng. Modified projection-type methods for monotone variational inequalities. *SIAM Journal on Control and Optimization*, 34(5):1814–1830, 1996. URL `citeseer.ist.psu.edu/article/solodov95modified.html`. 25

Georg Stadler. Semismooth newton and augmented lagrangian methods for a simplified friction problem. *SIAM Journal on Optimization*, 15(1):39–62, 2004. doi: 10.1137/S1052623403420833. URL `https://doi.org/10.1137/S1052623403420833`. 19

D.E. Stewart and J.C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction. *International Journal for Numerical Methods in Engineering*, 39(15), 1996. reference tiree du site WILEY. 7, 8

D. Sun and J. Sun. Strong semismoothness of the fischer-burmeister sdc and soc complementarity functions. *Mathematical Programming*, 103(3):575–581, Jul 2005. ISSN 1436-4646. doi: 10.1007/s10107-005-0577-4. URL `https://doi.org/10.1007/s10107-005-0577-4`. 27

A. Tasora and M. Anitescu. A fast NCP solver for large rigid-body problems with contacts, friction, and joints. In *Multibody dynamics*, volume 12 of *Comput. Methods Appl. Sci.*, pages 45–55. Springer, Berlin, 2009. 8, 38

A. Tasora and M. Anitescu. A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics. *Comput. Methods Appl. Mech. Engrg.*, 200(5-8):439–453, 2011. ISSN 0045-7825. doi: 10.1016/j.cma.2010.06.030. URL `http://dx.doi.org/10.1016/j.cma.2010.06.030`. 8, 38

A. Tasora and M. Anitescu. A complementarity-based rolling friction model for rigid contacts. *Meccanica*, 48(7):1643–1659, 2013. ISSN 0025-6455. doi: 10.1007/s11012-013-9694-y. URL `http://dx.doi.org/10.1007/s11012-013-9694-y`. 8

İ. Temizer, M.M. Abdalla, and Z. Gürdal. An interior point method for isogeometric contact. *Computer Methods in Applied Mechanics and Engineering*, 276(0):589 – 611, 2014. ISSN 0045-7825. doi: http://dx.doi.org/10.1016/j.cma.2014.03.018. URL `http://www.sciencedirect.com/science/article/pii/S0045782514001042`. 8

M.A. Tzaferopoulos. On an efficient new numerical method for the frictional contact problem of structures with convex energy density. *Computers & Structures*, 48(1):87–106, 1993. 22, 35

Xiang Wang, Bingsheng He, and Li-Zhi Liao. Steplengths in the extragradient type methods. *Journal of Computational and Applied Mathematics*, 233(11):2925 – 2939, 2010. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/j.cam.2009.11.037. URL `http://www.sciencedirect.com/science/article/pii/S0377042709007845`. 25

Barbara I. Wohlmuth and Rolf H. Krause. Monotone multigrid methods on nonmatching grids for nonlinear multibody contact problems. *SIAM Journal on Scientific Computing*, 25(1):324–347, 2003. doi: 10.1137/S1064827502405318. URL `https://doi.org/10.1137/S1064827502405318`. 8

P. Wriggers. *Computational Contact Mechanics*. Springer Verlag, second edition, 2006. originally published by John Wiley & Sons Ltd., 2002. 7, 12

L. Xuewen, A.-K. Soh, and C. Wanji. A new nonsmooth model for three-dimensional frictional contact problems. *Computational Mechanics*, 26:528–535, 2000. 19

Appendix 1. Basics in Convex Analysis

Definition 1 (Rockafellar and Wets [1997]). *Let $X \subseteq \mathbb{R}^n$. A multivalued (or point-to-set) mapping $T\colon X \rightrightarrows X$ is said to be (strictly) monotone if there exists $c(>) \geqslant 0$ such that for all $\hat{x}, \widetilde{x} \in X$*

$$(\hat{v} - \widetilde{v})^\top (\hat{x} - \widetilde{x}) \geqslant c\|\hat{x} - \widetilde{x}\| \qquad \text{with } \hat{v} \in T(\hat{x}), \widetilde{v} \in T(\widetilde{x}). \tag{137}$$

Moreover T is said to be maximal when it is not possible to add a pair (x, v) to the graph of T without destroying the monotonicity.

The Euclidean projector P_X onto a closed convex set X: for a vector $x \in \mathbb{R}^n$, the projected vector $z = P_X(x)$ is the unique solution of the convex quadratic programm

$$\begin{cases} \min \dfrac{1}{2}(y-x)^\top (y-x), \\ \text{s.t.} \quad y \in X. \end{cases} \tag{138}$$

The following equivalences are classical:

$$y = P_K(x) \iff \begin{array}{ll} \min & \frac{1}{2}(y-x)^\top (y-x) \\ \text{s.t.} & y \in K \end{array} \tag{139}$$

$$\iff -(y-x) \in N_K(y) \tag{140}$$

$$\iff (x-y)^\top (y-z) \geqslant 0, \forall z \in K \tag{141}$$

$$-F(x) \in N_K(x) \iff -\rho F(x)^\top (y-x) \geqslant 0, \forall y \in K \tag{142}$$

$$\iff (x - (x - \rho F(x))^\top (y-x) \geqslant 0, \forall y \in K \tag{143}$$

$$\iff x = P_K(x - \rho F(x)) \text{ thanks to (141)} \tag{144}$$

Sub-differential of the Euclidean norm.

The sub-differential of the Euclidean norm in $\mathbb{R}^n$ is given by:

$$\partial\|z\| = \begin{cases} \dfrac{z}{\|z\|}, & z \neq 0 \\ \{x, \|x\| \leqslant 1\}, & z = 0 \end{cases} \tag{145}$$

Euclidean projection on the unit ball.

Let $B = \{x \in \mathbb{R}^n, \|x\| \leqslant 1\}$. The Euclidean projection on the unit ball is given by:

$$P_B(z) = \begin{cases} z & \text{if} \quad z \in B \\ \dfrac{z}{\|z\|} & \text{if} \quad z \notin B \end{cases} \tag{146}$$

Its subdifferential can be computed as

$$\partial P_B(z) = \begin{cases} I & \text{if} \quad z \in B \setminus \partial B \\ I + (s-1)zz^\top, s \in [0,1] & \text{if} \quad z \in \partial B \\ \dfrac{I}{\|z\|} - \dfrac{zz^\top}{\|z\|^3} & \text{if} \quad z \notin B \end{cases} \tag{147}$$

Euclidean projection on the second order cone of $\mathbb{R}^3$.

Let $K = \{x = [x_{\mathrm{N}} x_{\mathrm{T}}]^T \in \mathbb{R}^3, x_{\mathrm{N}} \in \mathbb{R}, \|x_{\mathrm{T}}\| \leqslant \mu x_{\mathrm{N}}\}$ be the second order cone in $\mathbb{R}^3$. The Euclidean projection on K is

$$P_K(z) = \begin{cases} z & \text{if} \quad z \in K \\ 0 & \text{if} \quad -z \in K^* \\ \dfrac{1}{1+\mu^2}(z_{\mathrm{N}} + \mu\|z_{\mathrm{T}}\|) \begin{bmatrix} 1 \\ \mu \dfrac{z_{\mathrm{T}}}{\|z_{\mathrm{T}}\|} \end{bmatrix} & \text{if} \quad z \notin K \text{ and } -z \notin K^* \end{cases} \tag{148}$$

Direct computation of an element of the subdifferential The computation of the subdifferential of P_K is given as follows

- if $z \in K \setminus \partial K$, $\partial_z P_K(z) = I$,
- if $-z \in K^* \setminus \partial K^*$, $\partial_z P_K(z) = 0$,
- if $z \notin K$ and $-z \notin K^*$ and, $\partial_z P_K(z) = 0$, we get

$$\partial_{z_{\mathrm{N}}} P_K(z) = \frac{1}{1+\mu^2} \begin{bmatrix} 1 \\ \mu z_{\mathrm{T}} \end{bmatrix} \tag{149}$$

and

$$\partial_{z_{\mathrm{T}}} [P_K(z)]_{\mathrm{N}} = \frac{\mu}{1+\mu^2} \frac{z_{\mathrm{T}}}{\|z_{\mathrm{T}}\|} \tag{150}$$

$$\partial_{z_{\mathrm{T}}} [P_K(z)]_{\mathrm{T}} = \frac{\mu}{(1+\mu^2)} \left[\mu \frac{z_{\mathrm{T}}}{\|z_{\mathrm{T}}\|} \frac{z_{\mathrm{T}}^\top}{\|z_{\mathrm{T}}\|} + (z_{\mathrm{N}} + \mu\|z_{\mathrm{T}}\|)\left(\frac{I_2}{\|z_{\mathrm{T}}\|} - \frac{z_{\mathrm{T}} z_{\mathrm{T}}^\top}{\|z_{\mathrm{T}}\|^3}\right) \right] \tag{151}$$

that is

$$\partial_{z_{\mathrm{T}}} [P_K(z)]_{\mathrm{T}} = \frac{\mu}{(1+\mu^2)\|z_{\mathrm{T}}\|} \left[(z_{\mathrm{N}} + \mu\|z_{\mathrm{T}}\|)\, I_2 + z_{\mathrm{N}} \frac{z_{\mathrm{T}} z_{\mathrm{T}}^\top}{\|z_{\mathrm{T}}\|^2}) \right] \tag{152}$$

Computation of the subdifferential using the spectral decomposition In [Hayashi et al., 2005], the computation of the Clarke subdifferential of the projection operator is also done by inspecting the different cases using the spectral decomposition

$$\partial P_K(x) = \begin{cases} I & (\lambda_1 > 0, \lambda_2 > 0) \\ \dfrac{\lambda_2}{\lambda_1 + \lambda_2} I + Z & (\lambda_1 < 0, \lambda_2 > 0) \\ 0 & (\lambda_1 < 0, \lambda_2 < 0) \\ \mathrm{co}\{I, I + Z\} & (\lambda_1 = 0, \lambda_2 > 0) \\ \mathrm{co}\{0, Z\} & (\lambda_1 < 0, \lambda_2 = 0) \\ \mathrm{co}\{0 \cup I \cup S\} & (\lambda_1 = 0, \lambda_2 = 0) \end{cases} \tag{153}$$

where

$$\begin{aligned} Z &= \tfrac{1}{2} \begin{bmatrix} -y_{\mathrm{N}} & y_{\mathrm{T}}^\top \\ y_{\mathrm{T}} & -y_{\mathrm{N}} y_{\mathrm{T}} y_{\mathrm{T}}^\top \end{bmatrix}, \\ S &= \left\{ \tfrac{1}{2}(1+\beta) I + \tfrac{1}{2} \begin{bmatrix} -\beta & w^\top \\ w & -\beta w w^\top \end{bmatrix} \mid -1 \leqslant \beta \leqslant 1, \|w\| = 1 \right\} \end{aligned} \tag{154}$$

with $y = x/\|x_{\mathrm{T}}\|$. A simple verification shows that the previous computation is an element of the subdifferential.

Appendix 2. Computation of Generalized Jacobians for Nonsmooth Newton methods

Computation of components of subgradient of $F_{\text{vi}}^{\mathsf{nat}}$

Let us introduce the following notation for an element of the sub–differential

$$\Phi(u,r) = \left[\begin{array}{cc} \rho I & -\rho W \\ \Phi_{ru}(u,r) & \Phi_{rr}(u,r) \end{array} \right] \in \partial F_{\text{vi}}^{\mathsf{nat}}(u,r) \tag{155}$$

where $\Phi_{xy}(u,r) \in \partial_x [F_{\text{vi}}^{\mathsf{nat}}]_y(u,r)$. Since $\Phi_{uu}(u,r) = I$, a reduction of the system is performed in practice and Algorithm 4 is applied or $z = r$ with

$$\begin{cases} G(z) = [F_{\text{vi}}^{\mathsf{nat}}]_r(Wr+q,r) \\ \Phi(z) = \Phi_{rr}(r,Wr+q) + \Phi_{ru}(r,Wr+q)W \end{cases} \tag{156}$$

Let us introduce the following notation for an element of the sub–differential with an obvious simplification

$$\Phi(v,r) = \left[\begin{array}{ccc} \rho M & -\rho H & \\ -\rho H^\top & \rho I & 0 \\ 0 & \Phi_{ru}(v,u,r) & \Phi_{rr}(v,u,r) \end{array} \right] \in \partial F_{\text{vi}}^{\mathsf{nat}}(u,r) \tag{157}$$

where $\Phi_{xy}(v,u,r) \in \partial_x [F_{\text{vi-1}}^{\mathsf{nat}}]_y(v,u,r)$. A possible computation of $\Phi_{ru}(v,u,r)$ and $\Phi_{rr}(v,u,r)$ is directly given by (159) and (158). In this case, the variable u can be also substituted.

For one contact, a possible computation of the remaining parts in $\Phi(u,r)$ is given by

$$\Phi_{ru}(u,r) = \begin{cases} 0 & \text{if} \quad r - \rho(u + g(u)) \in K \\ \\ I - \partial_r [P_K(r - \rho(u+g(u)))] & \text{if} \quad r - \rho(u+g(u)) \notin K \end{cases} \tag{158}$$

$$\Phi_{ru}(u,r) = \begin{cases} \rho \left(I + \left[\begin{array}{ccc} 0 & 0 & 0 \\ \dfrac{u_{\text{T}}}{\|u_{\text{T}}\|} & 0 & 0 \end{array} \right] \right) & \text{if} \quad \begin{cases} r - \rho(u+g(u)) \in K \\ u_{\text{T}} \neq 0 \end{cases} \\ \\ \rho \left(I + \left[\begin{array}{ccc} 0 & 0 & 0 \\ s & 0 & 0 \end{array} \right] \right), s \in \mathbb{R}^2, \|s\| = 1 & \text{if} \quad \begin{cases} r - \rho(u+g(u)) \in K \\ u_{\text{T}} = 0 \end{cases} \\ \\ I + \rho \left(I + \left[\begin{array}{ccc} 0 & 0 & 0 \\ \dfrac{u_{\text{T}}}{\|u_{\text{T}}\|} & 0 & 0 \end{array} \right] \right) \partial_u [P_K(r - \rho(u+g(u)))] & \text{if} \quad r - \rho(u+g(u)) \notin K \end{cases} \tag{159}$$

The computation of an element of ∂P_K is given in Appendix 11.

Alart–Curnier function and its variants

For one contact, a possible computation of the remaining parts in $\Phi(u,r)$ is given by

$$\Phi_{r_{\text{N}} u_{\text{N}}}(u,r) = \begin{cases} \rho_{\text{N}} & \text{if } r_{\text{N}} - \rho_{\text{N}} u_{\text{N}} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{160}$$

$$\Phi_{r_{\mathrm{N}} r_{\mathrm{N}}}(u,r) = \begin{cases} 0 & \text{if } r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}} > 0 \\ 1 & \text{otherwise} \end{cases} \tag{161}$$

$$\Phi_{r_{\mathrm{T}} u_{\mathrm{N}}}(u,r) = \begin{cases} 0 & \text{if } \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| \leqslant \mu \max(0, r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) \\ 0 & \text{if } \begin{cases} \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| > \mu \max(0, r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) \\ r_{\mathrm{N}} - \rho_{\mathrm{N}} u_n \leqslant 0 \end{cases} \\ \mu \rho_{\mathrm{N}} \dfrac{r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}}{\|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\|} & \text{if } \begin{cases} \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| > \mu \max(0, r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) \\ r_{\mathrm{N}} - \rho_{\mathrm{N}} u_n > 0 \end{cases} \end{cases} \tag{162}$$

$$\Phi_{r_{\mathrm{T}} u_{\mathrm{T}}}(u,r) = \begin{cases} \rho_{\mathrm{T}} & \text{if } \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| \leqslant \mu \max(0, r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) \\ \mu \rho_{\mathrm{T}} (r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}})_+ \Gamma(r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}) & \text{if } \begin{cases} \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| > \mu \max(0, r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) \\ r_{\mathrm{N}} - \rho_{\mathrm{N}} u_n > 0 \end{cases} \end{cases} \tag{163}$$

$$\Phi_{r_{\mathrm{T}} r_{\mathrm{N}}}(u,r) = \begin{cases} 0 & \text{if } \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| \leqslant \mu \max(0, r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) \\ 0 & \text{if } \begin{cases} \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| > \mu \max(0, r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) \\ r_{\mathrm{N}} - \rho_{\mathrm{N}} u_n \leqslant 0 \end{cases} \\ -\mu \dfrac{r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}}{\|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\|} & \text{if } \begin{cases} \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| > \mu \max(0, r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) \\ r_{\mathrm{N}} - \rho_{\mathrm{N}} u_n > 0 \end{cases} \end{cases} \tag{164}$$

$$\Phi_{r_{\mathrm{T}} r_{\mathrm{T}}}(u,r) = \begin{cases} 0 & \text{if } \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| \leqslant \mu \max(0, r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) \\ I_2 - \mu (r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}})_+ \Gamma(r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}) & \text{if } \begin{cases} \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| > \mu \max(0, r_{\mathrm{N}} - \rho_{\mathrm{N}} u_{\mathrm{N}}) \\ r_{\mathrm{N}} - \rho_{\mathrm{N}} u_n > 0 \end{cases} \end{cases} \tag{165}$$

with the function $\Gamma(\cdot)$ defined by

$$\Gamma(x) = \frac{I_{2\times 2}}{\|x\|} - \frac{x\, x^\top}{\|x\|^3} \tag{166}$$

If the variant (60) is chosen, the computation of $\Phi_{r_{\mathrm{T}} \bullet}$ simplifies in

$$\Phi_{r_{\mathrm{T}} u_{\mathrm{N}}}(u,r) = 0 \tag{167}$$

$$\Phi_{r_{\mathrm{T}} u_{\mathrm{T}}}(u,r) = \begin{cases} \rho_{\mathrm{T}} & \text{if } \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| \leqslant \mu r_{\mathrm{N}} \\ -\mu \rho_{\mathrm{T}} r_{n,+} \Gamma(r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}) & \text{if } \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| > \mu r_{\mathrm{N}} \end{cases} \tag{168}$$

$$\Phi_{r_{\mathrm{T}} r_{\mathrm{N}}}(u,r) = \begin{cases} 0 & \text{if } \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| \leqslant \mu r_{\mathrm{N}} \\ 0 & \text{if } \begin{cases} \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| > \mu r_n \\ r_{\mathrm{N}} \leqslant 0 \end{cases} \\ -\mu \dfrac{r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}}{\|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\|} & \text{if } \begin{cases} \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| > \mu r_n \\ r_{\mathrm{N}} > 0 \end{cases} \end{cases} \tag{169}$$

$$\Phi_{r_{\mathrm{T}} r_{\mathrm{T}}}(u,r) = \begin{cases} 0 & \text{if } \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| \leqslant \mu r_{\mathrm{N}} \\ I_2 - \mu (r_{\mathrm{N}})_+ \Gamma(r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}) & \text{if } \|r_{\mathrm{T}} - \rho_{\mathrm{T}} u_{\mathrm{T}}\| > \mu r_{\mathrm{N}} \end{cases} \tag{170}$$

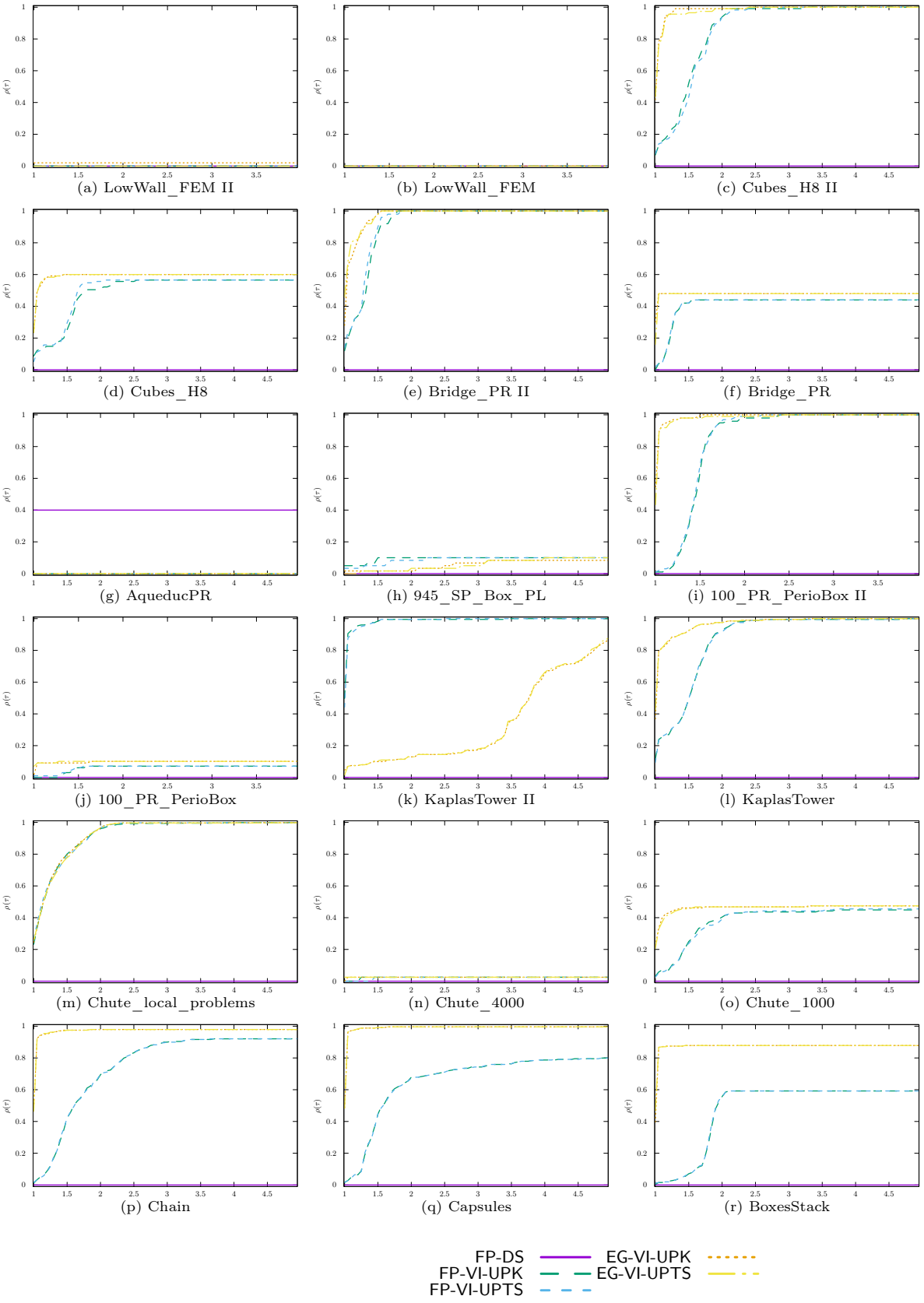


Figure 16: Comparison of numerical method for VI FP-DS, FP-VI-⋆ and FP-EG-⋆

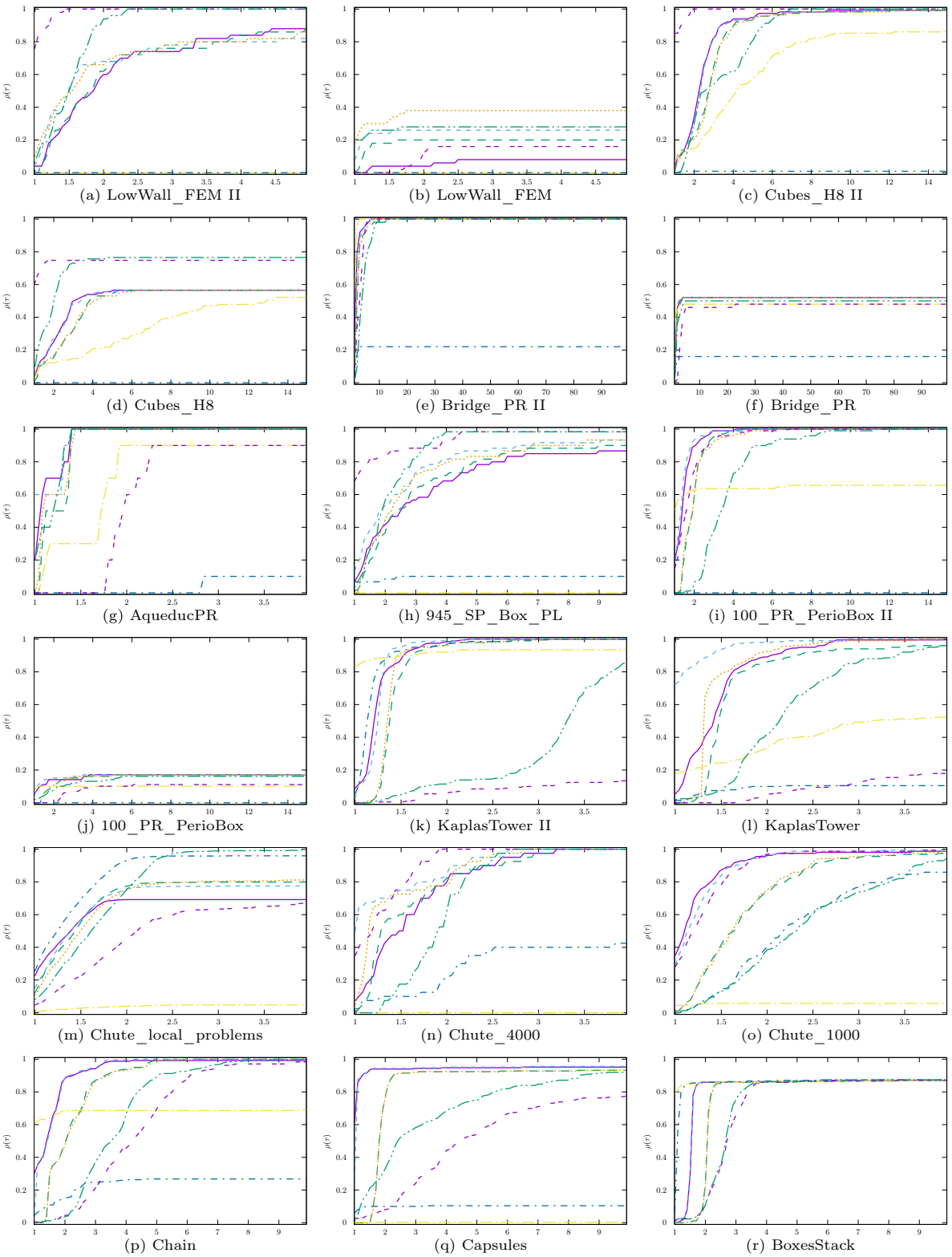


Figure 17: Influence of the local solver in NSGS-⋆ algorithms.

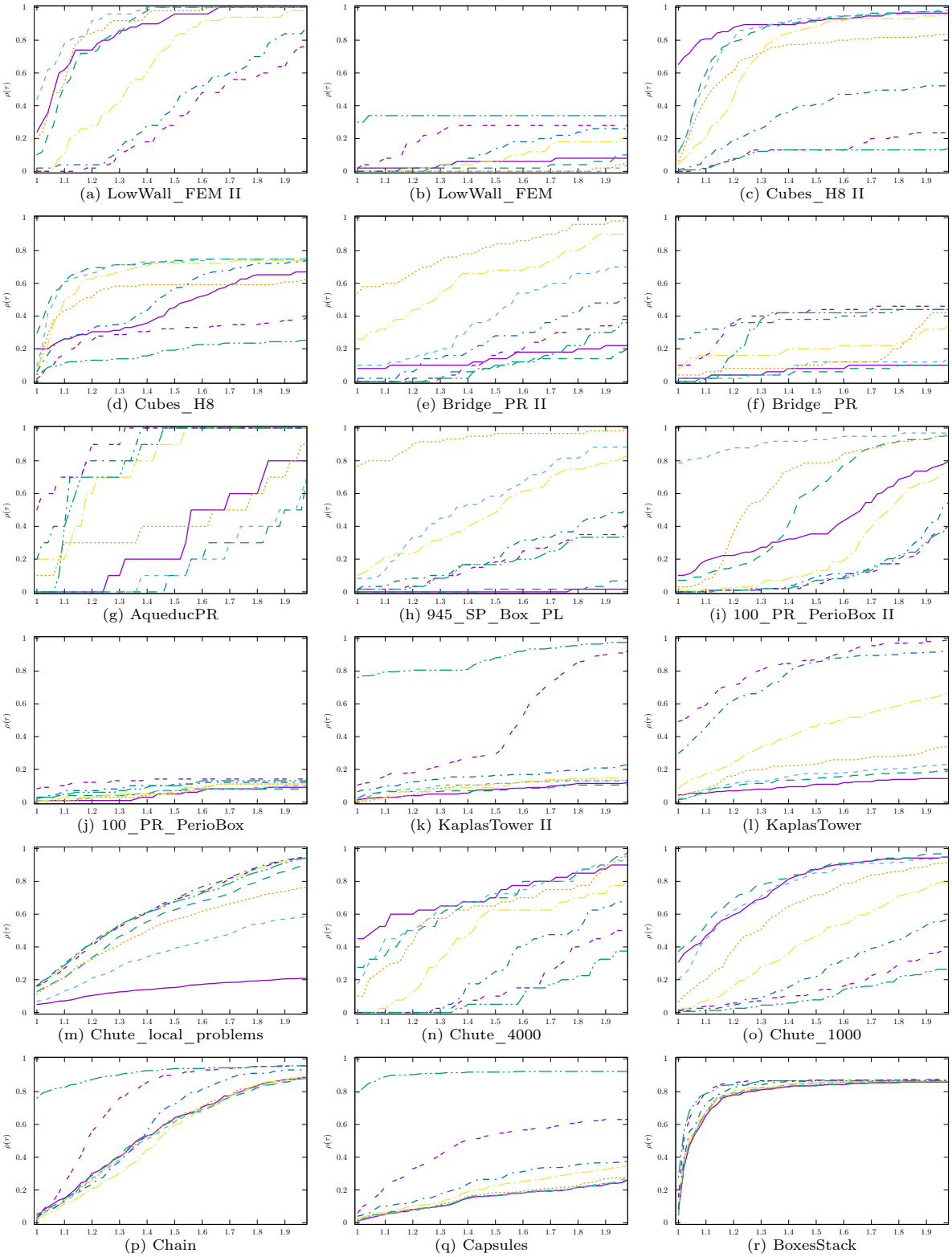


Figure 18: Influence of the tolerance of the local solver $\mathsf{tol}_{\mathsf{local}}$ in NSGS-FP-VI-UPK algorithms.

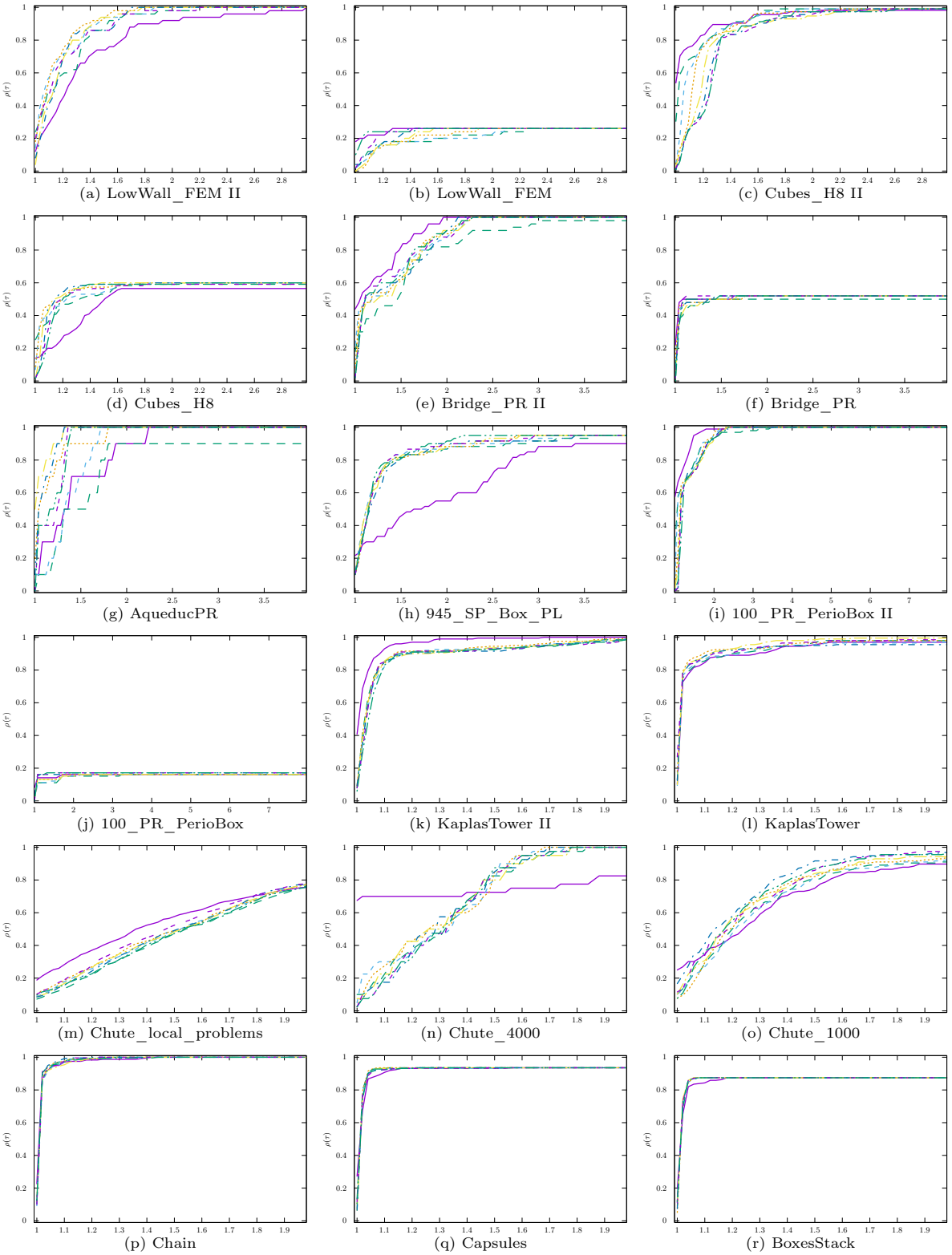


Figure 19: Influence of the tolerance of the local solver $\mathsf{tol}_{\mathsf{local}}$ in NSGS-FP-NSN-AC-GP algorithms.

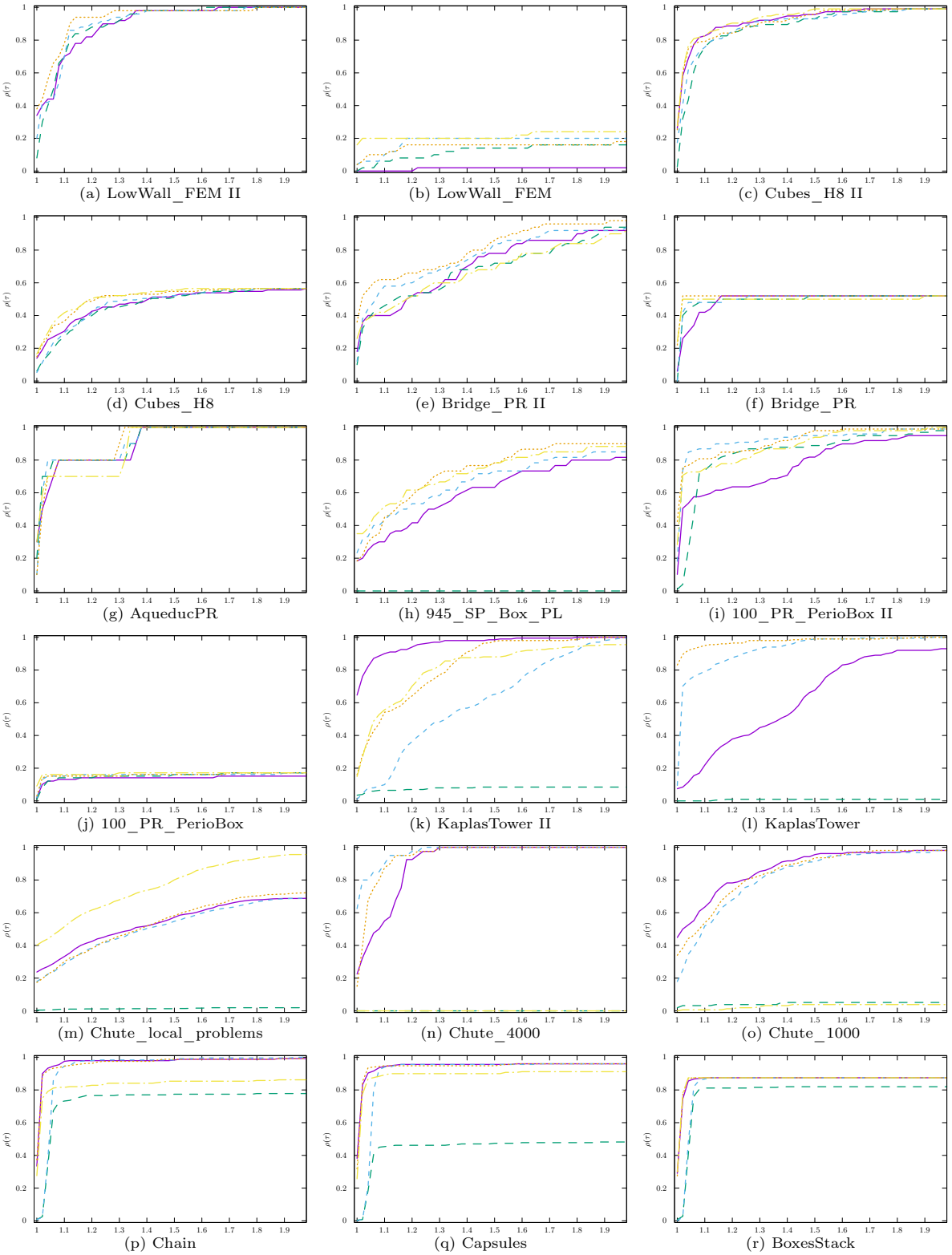


Figure 20: Influence of the choice of the parameters $\rho_{\mathrm{N}}, \rho_{\mathrm{T}}$ in the local solver of the NSGS-AC algorithms

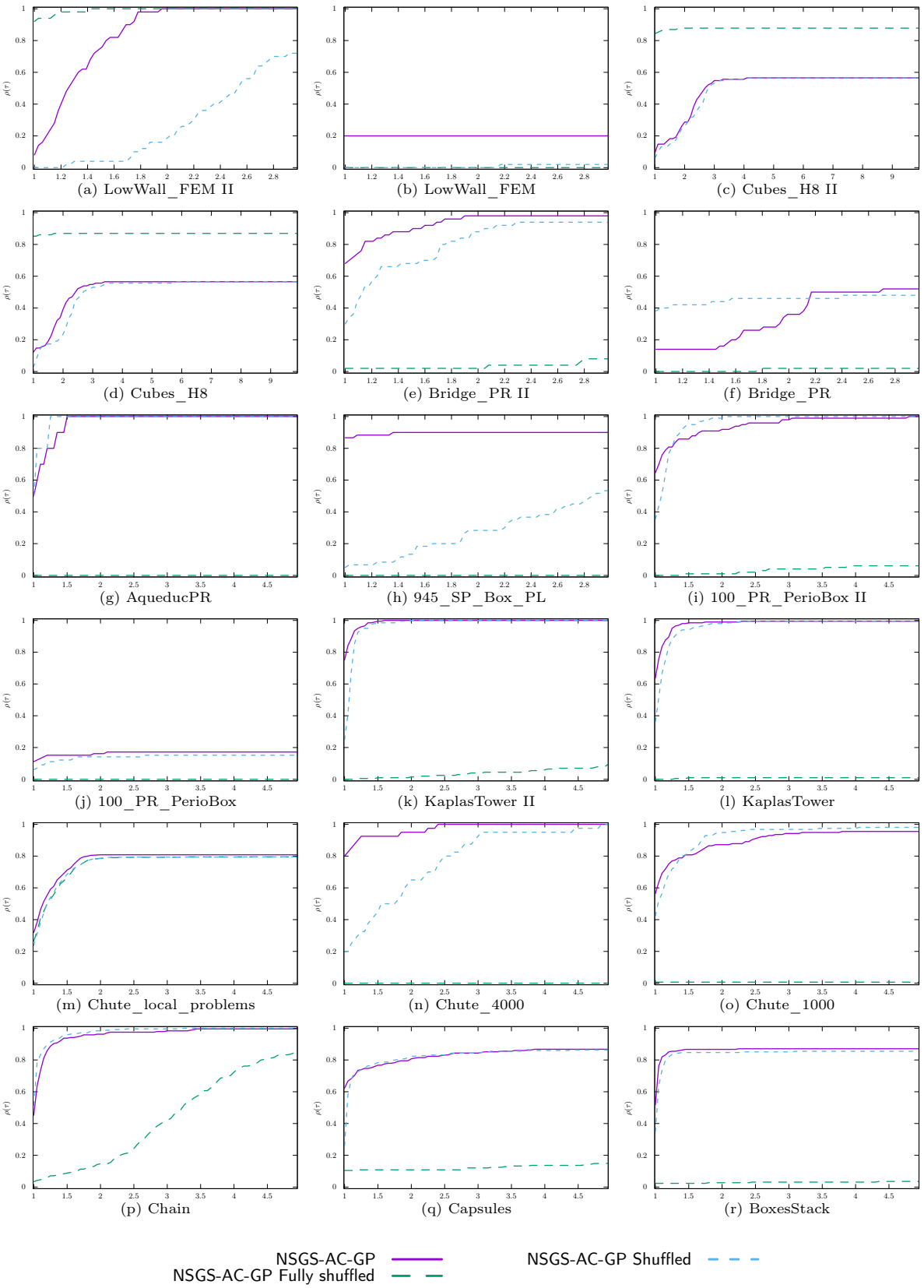


Figure 21: Influence of the contacts order in NSGS algorithms.

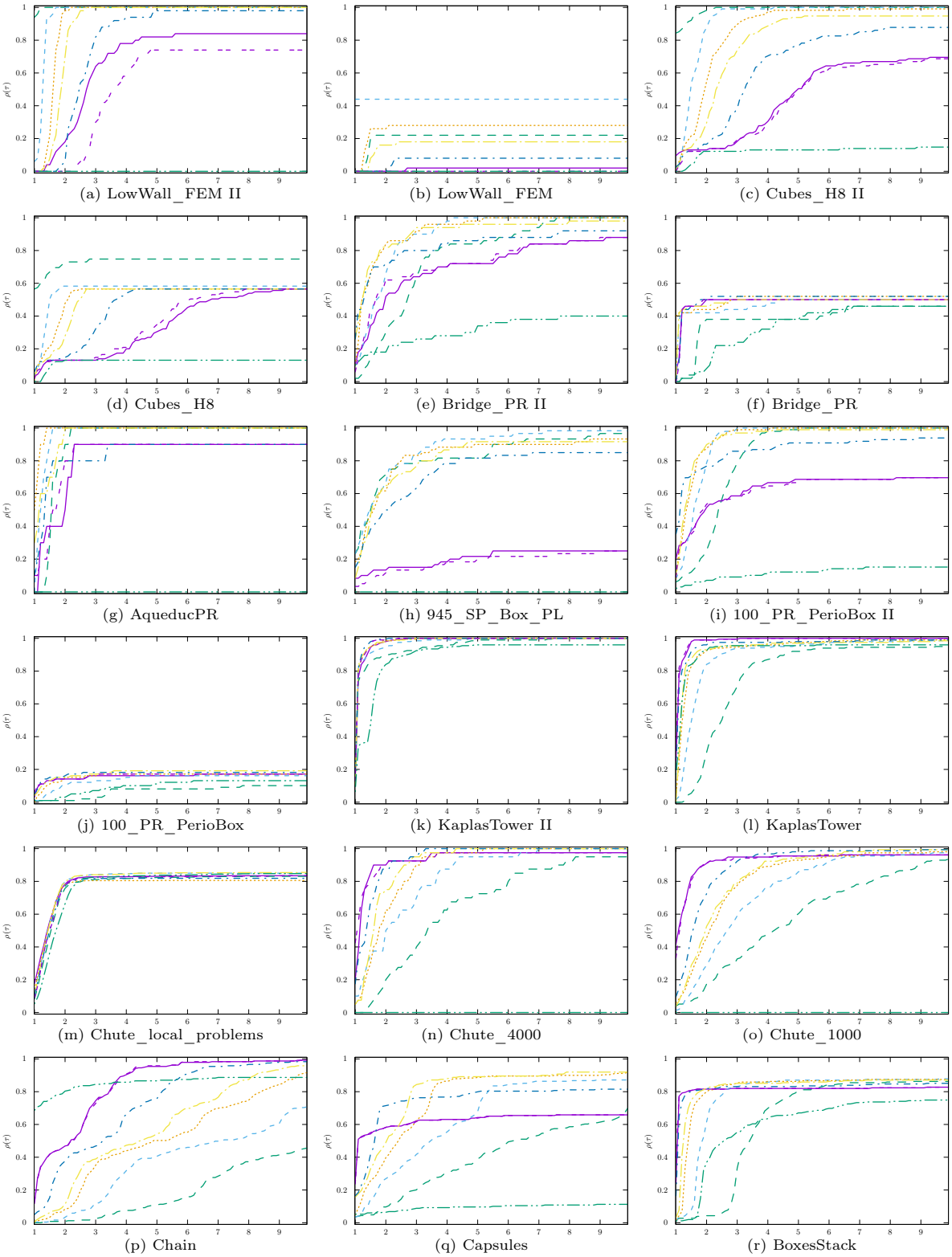


Figure 22: Effect of relation coefficient ω in PSOR-AC-GP algorithm.

Appendix 3. Full report of tests

Numerical methods for VI: FP-DS, FP-VI-$\star$ **and** FP-EG-$\star$

Splitting based algorithms: NSGS-$\star$ **and** PSOR-$\star$

Comparison of NSN-$\star$ **algorithms**

Comparison of PPA-NSN-AC algorithm with respect to the step-size parameter σ, μ

Comparison of optimization-based algorithms

Comparison of different families of solvers.

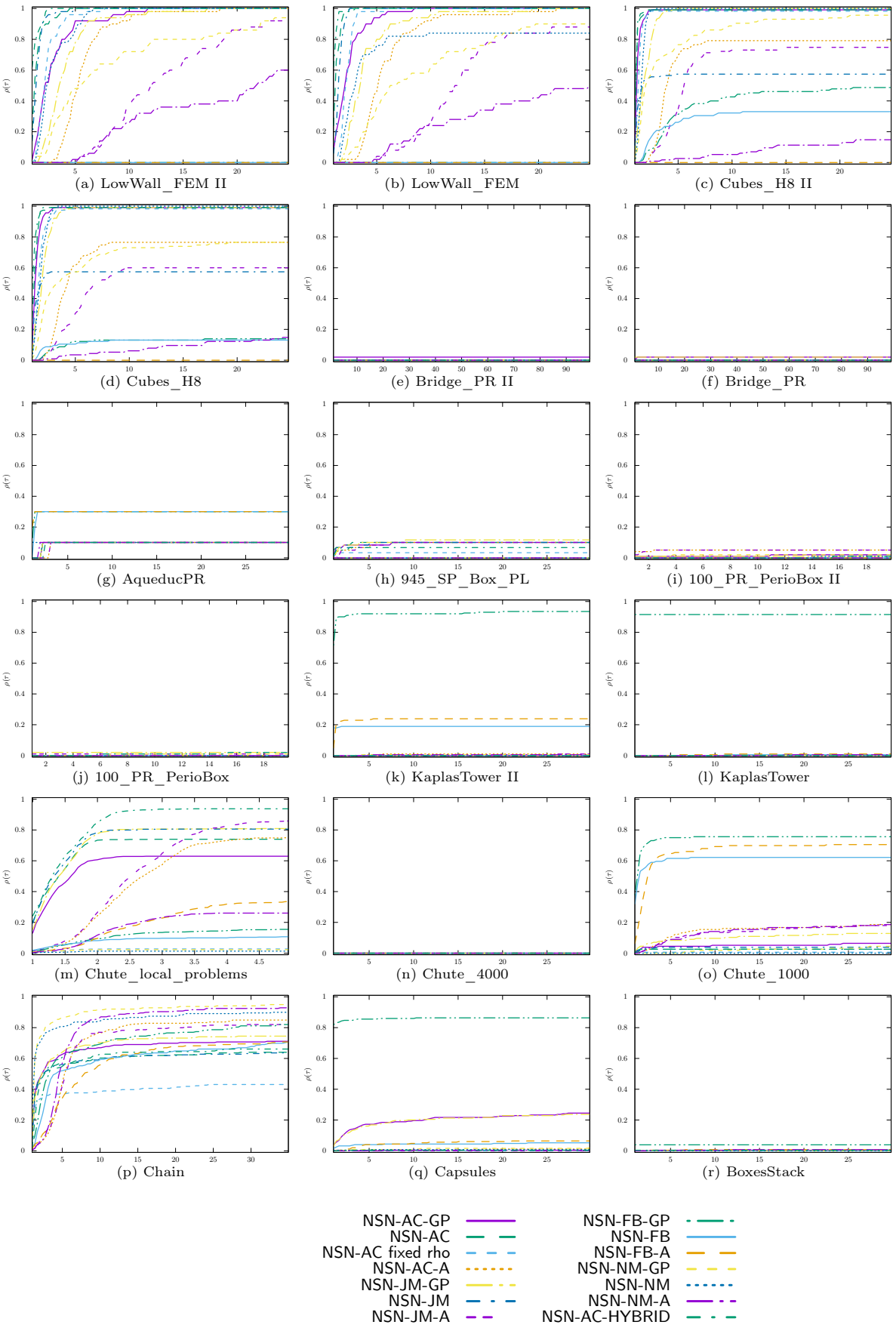


Figure 23: Comparison of NSN-⋆ algorithms.

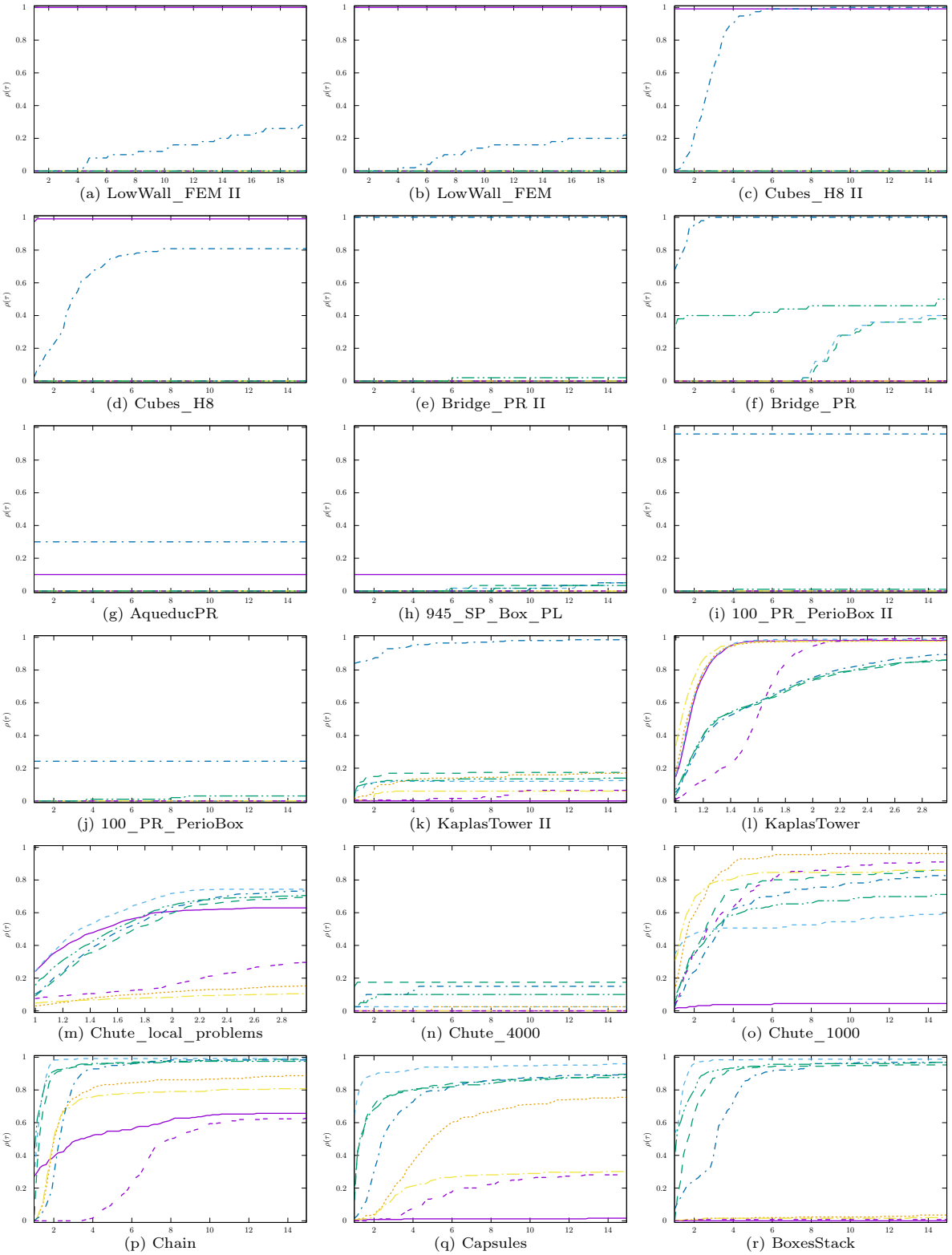


Figure 24: Comparison of internal solvers in PPA-⋆ algorithms.

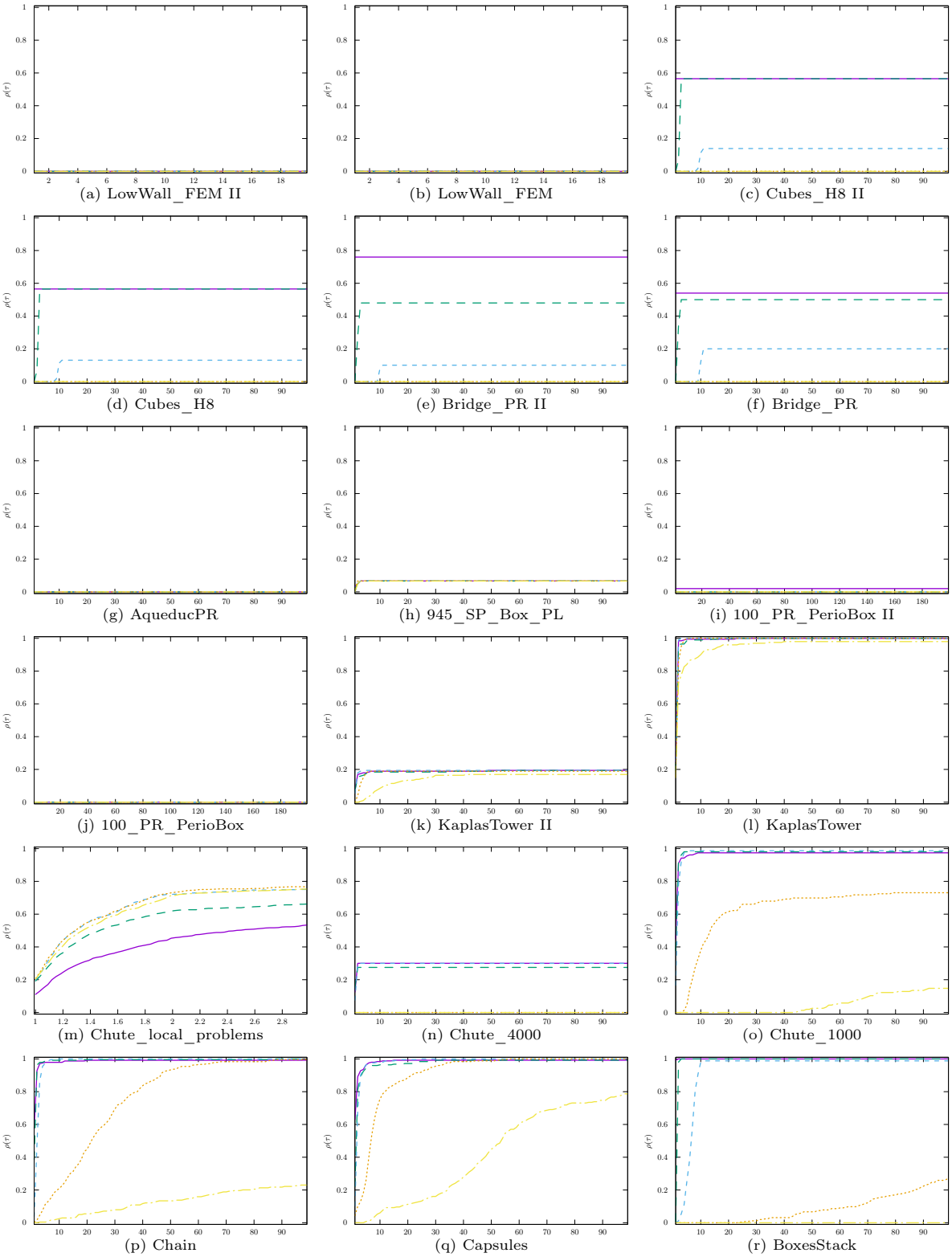


Figure 25: Effect of the step-size parameter σ, μ in PPA-NSN-AC algorithm

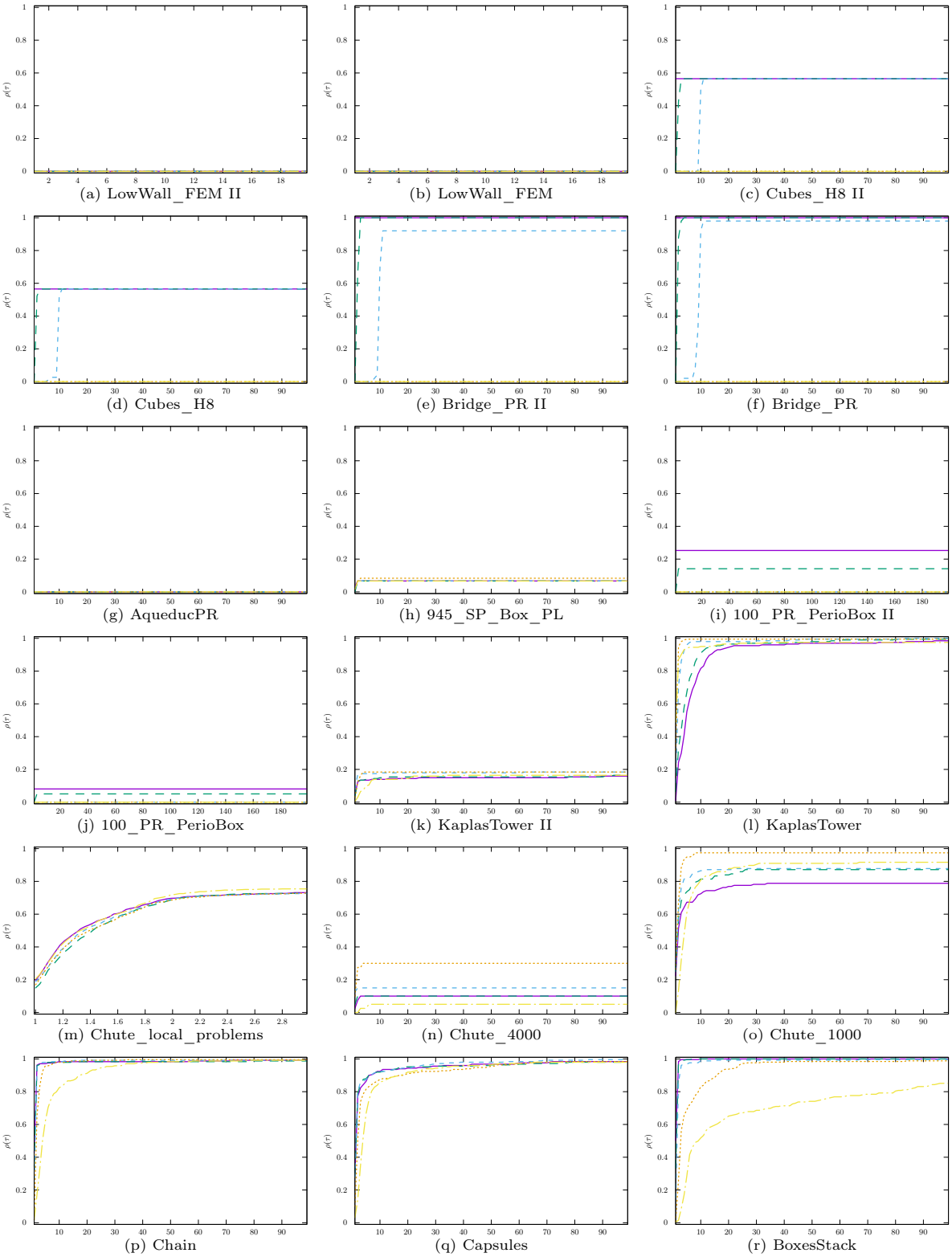


Figure 26: Effect of the step-size parameter σ, μ in PPA-NSN-AC algorithm

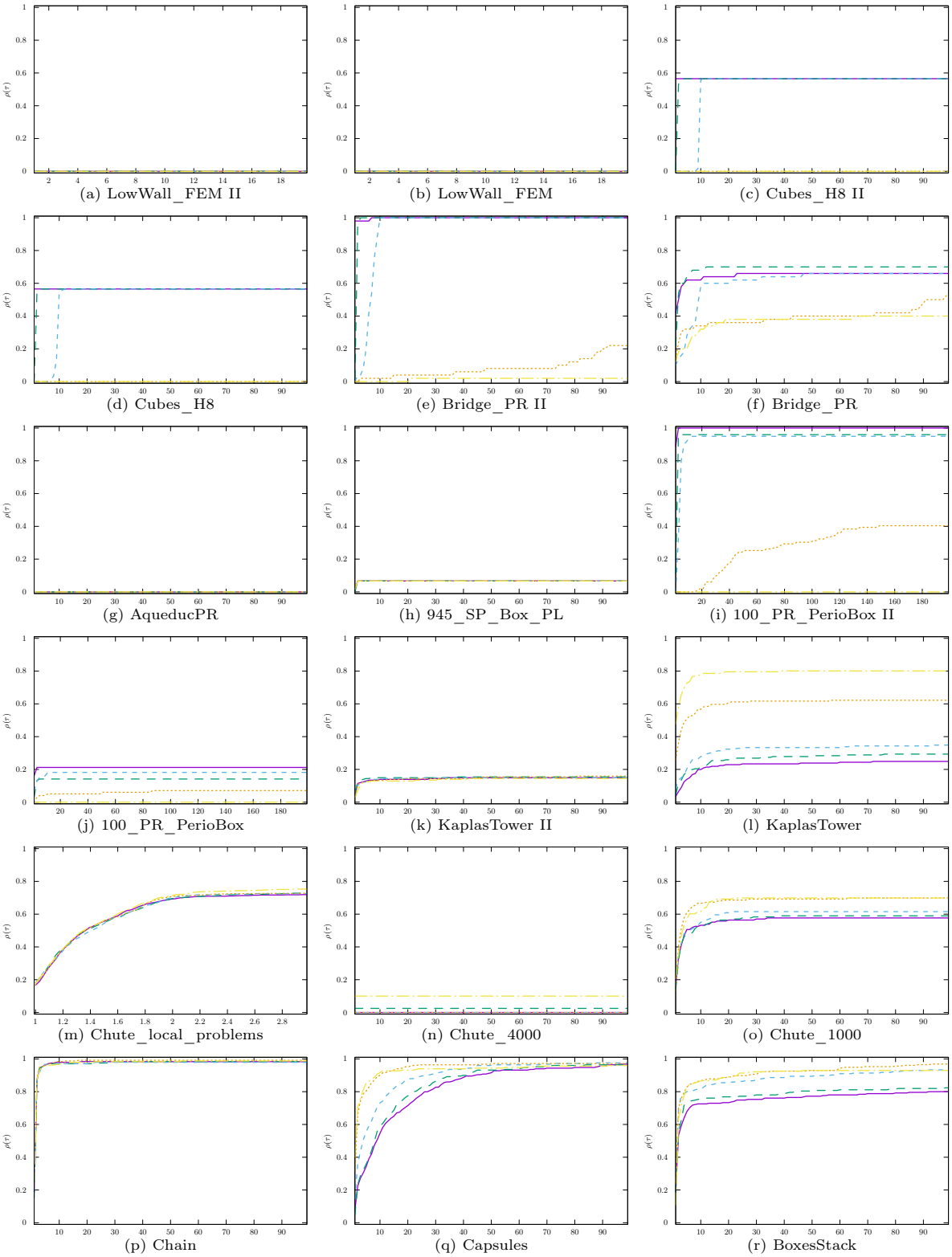


Figure 27: Effect of the step-size parameter σ, μ in PPA-NSN-AC algorithm

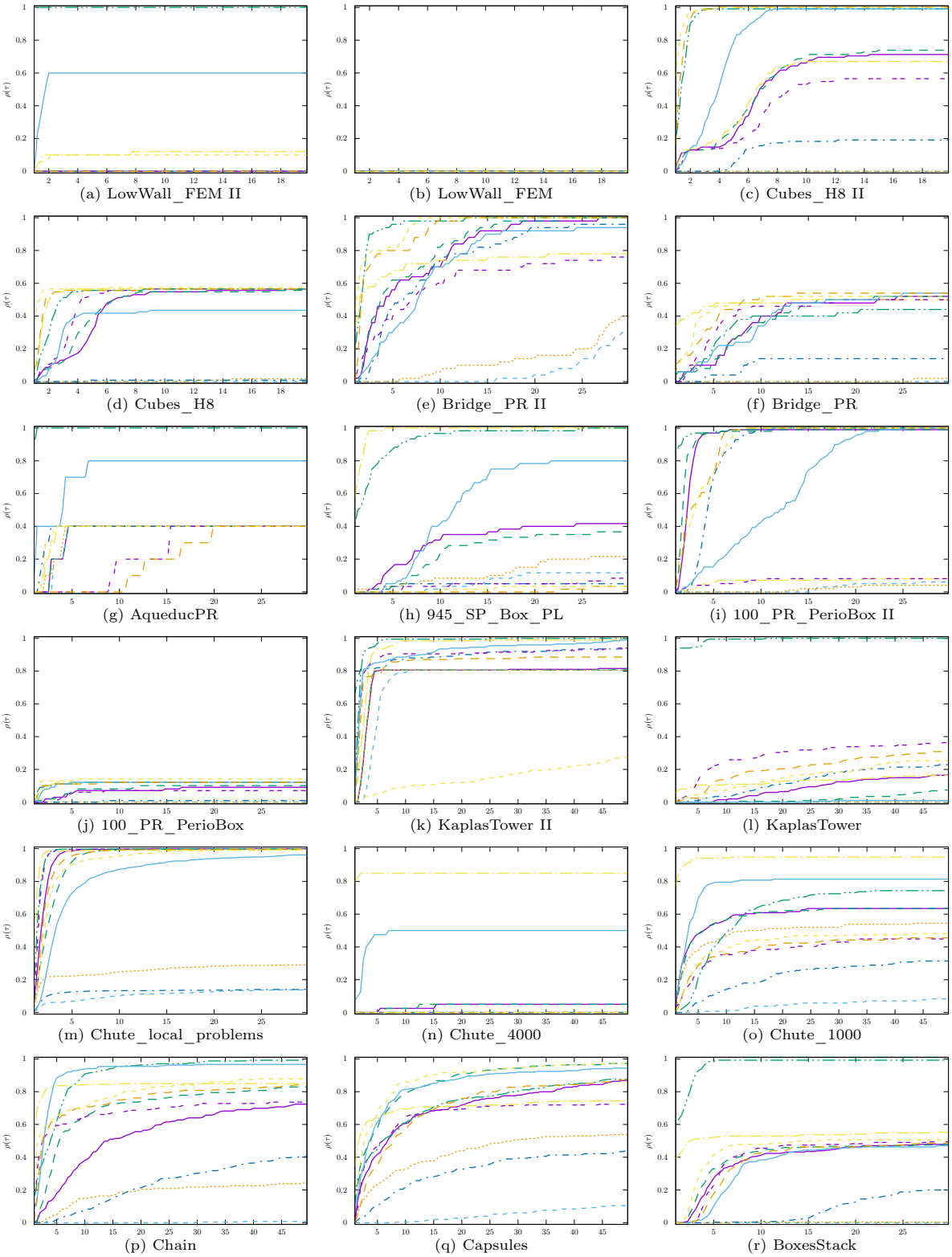


Figure 28: Comparison of the optimization based solvers

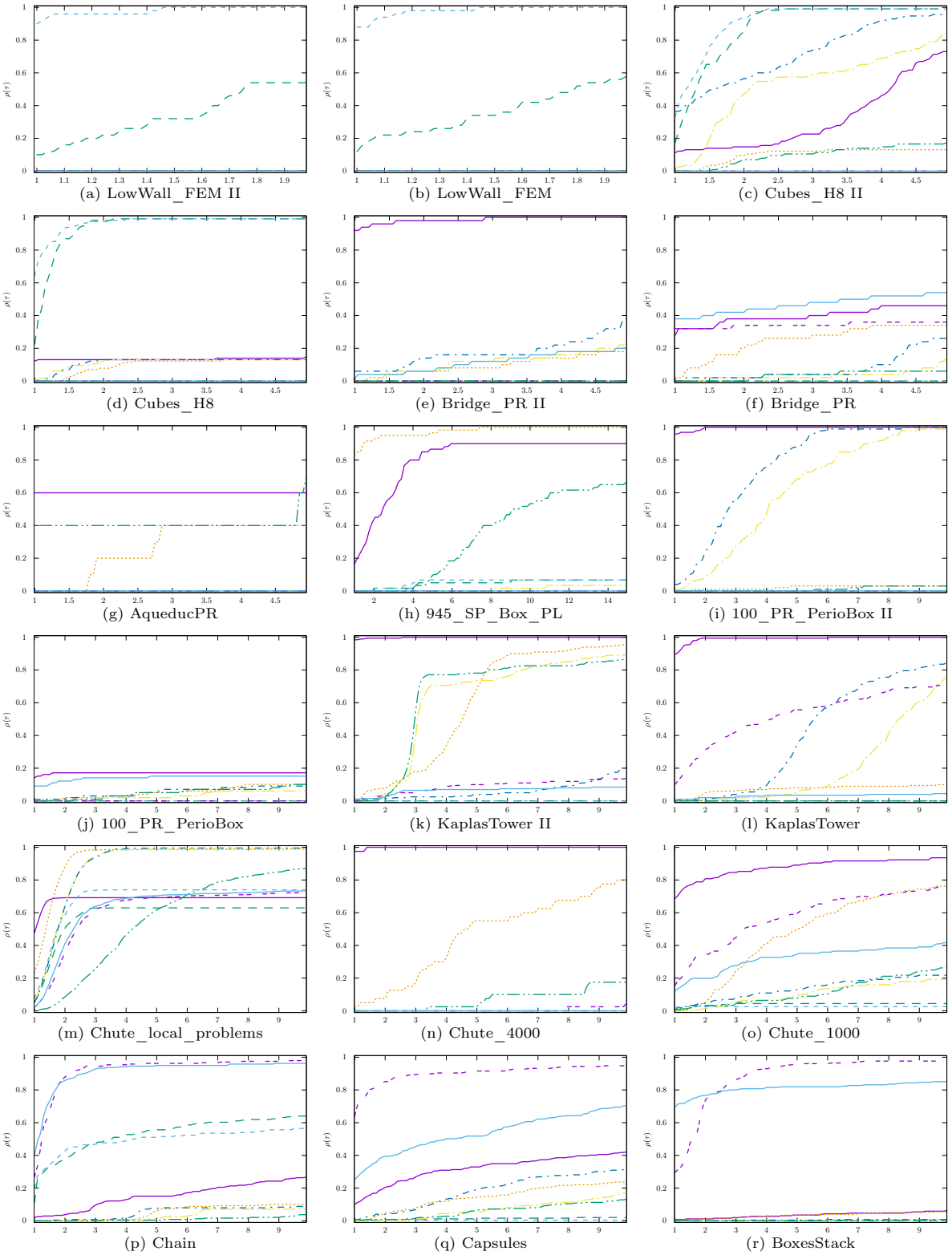


Figure 29: Comparison of the solvers between families

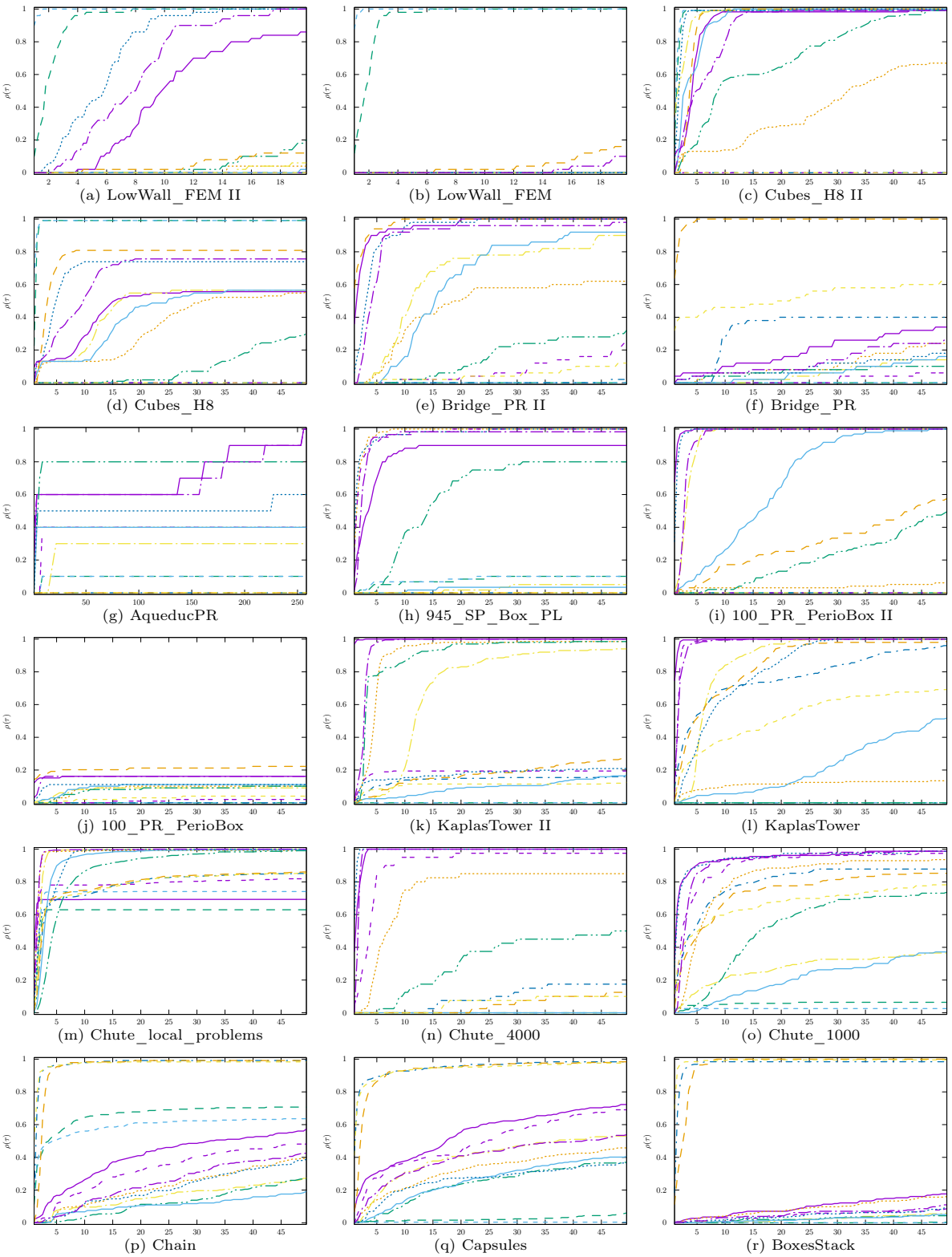


Figure 30: Comparison of the solvers between families

LMGC_100_PR_PerioBox precision 1.0e-04 timeout 100

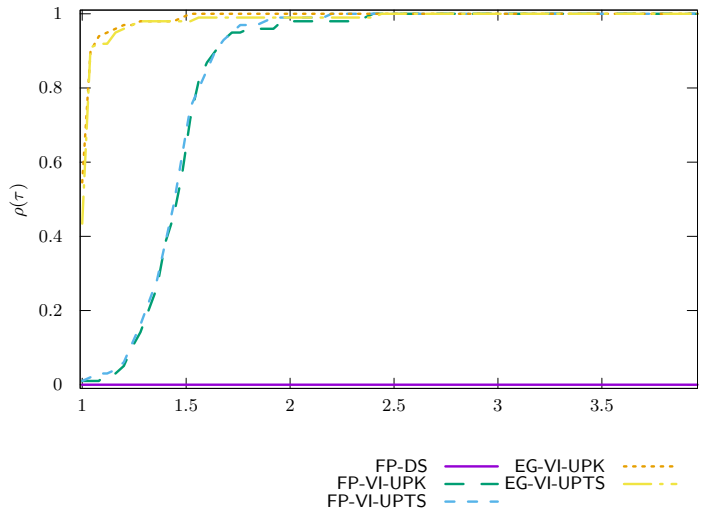


Figure 31: LMGC_100_PR_PerioBox time VI/UpdateRule

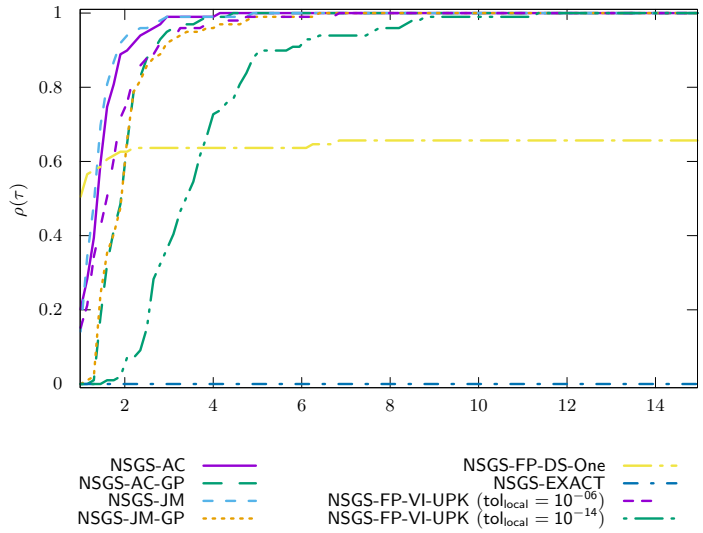


Figure 32: LMGC_100_PR_PerioBox time NSGS/LocalSolver

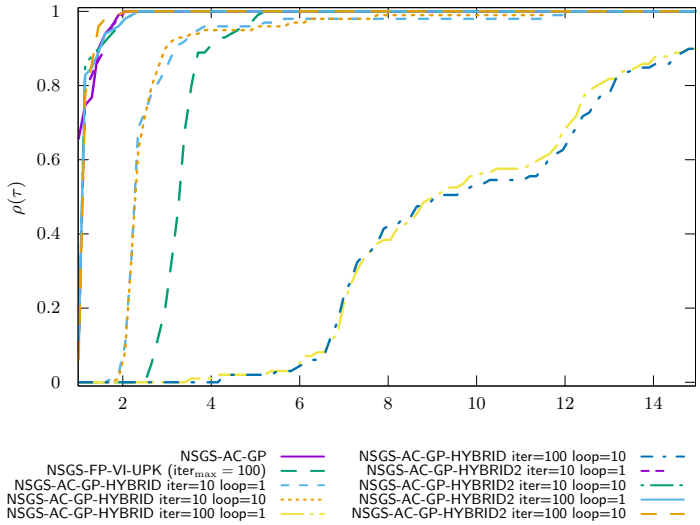


Figure 33: LMGC_100_PR_PerioBox time NSGS/LocalSolverHybrid

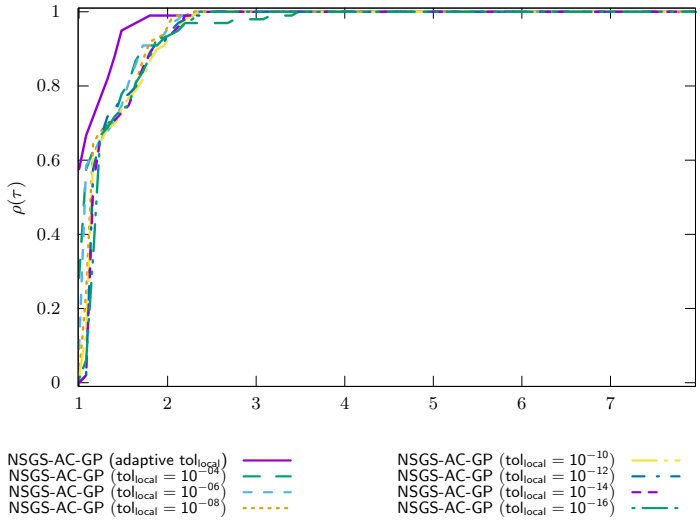


Figure 34: LMGC_100_PR_PerioBox time NSGS/LocalTol

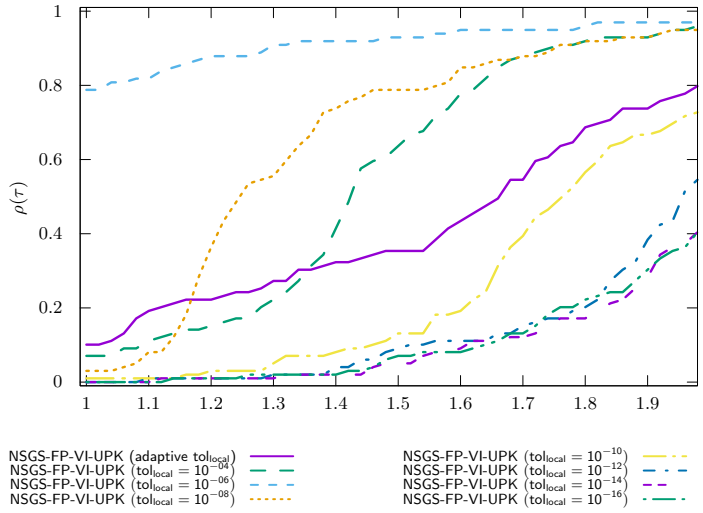


Figure 35: LMGC_100_PR_PerioBox time NSGS/LocalTol-VI

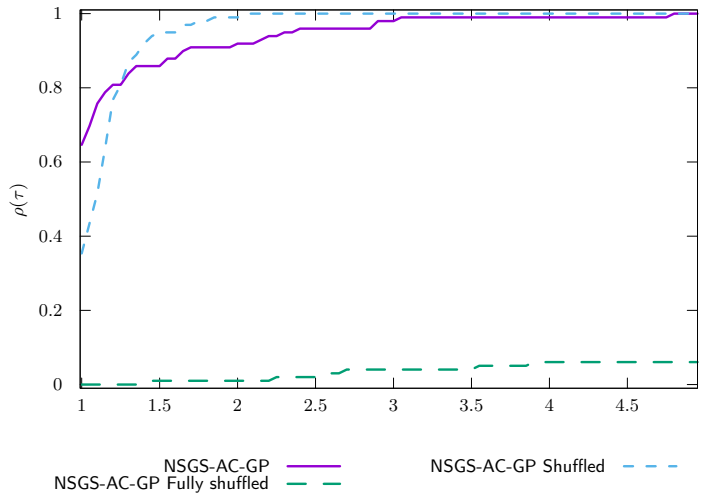


Figure 36: LMGC_100_PR_PerioBox time NSGS/Shuffled

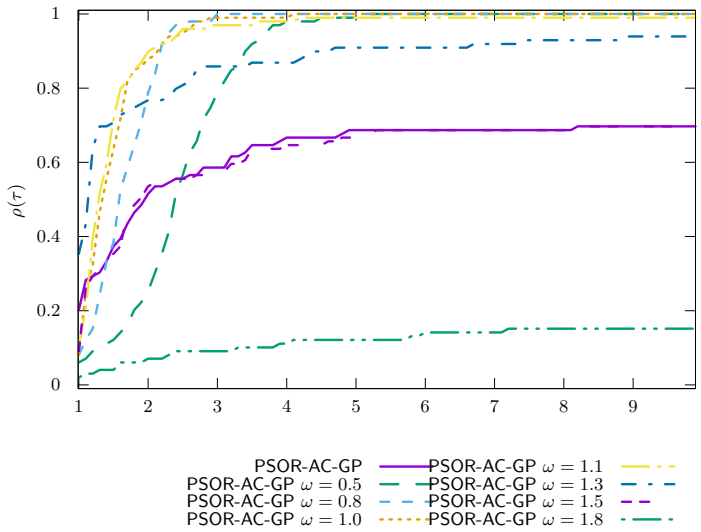


Figure 37: LMGC_100_PR_PerioBox time PSOR

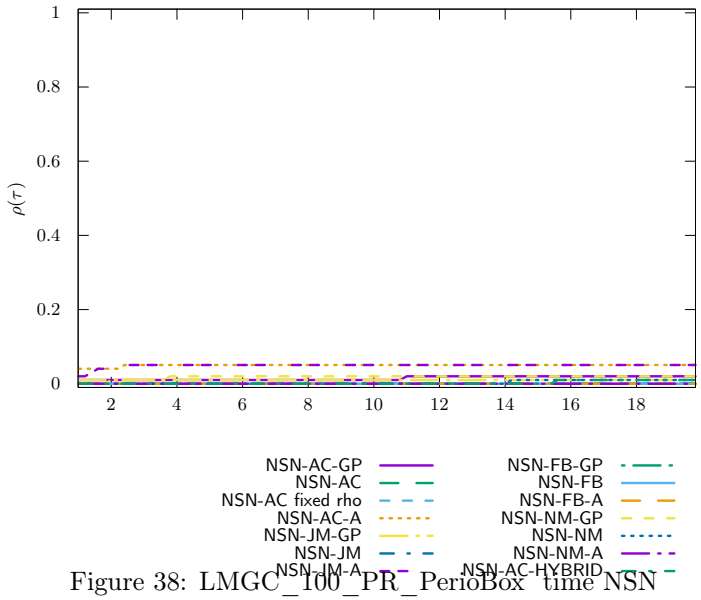


Figure 38: LMGC_100_PR_PerioBox time NSN

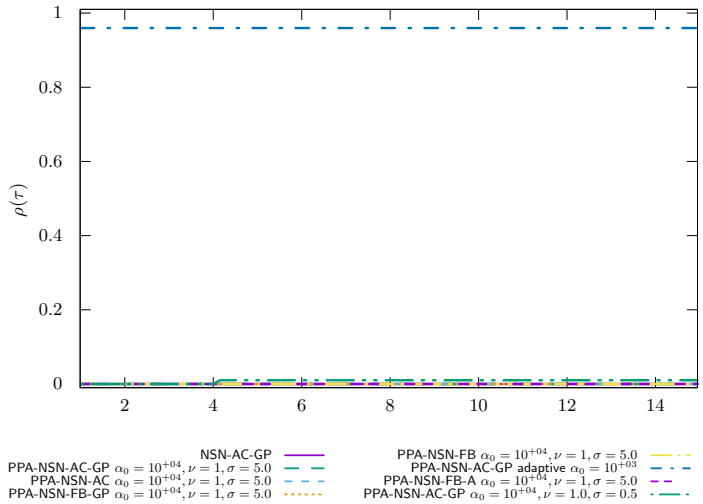


Figure 39: LMGC_100_PR_PerioBox time PROX/NSN/InternalSolvers

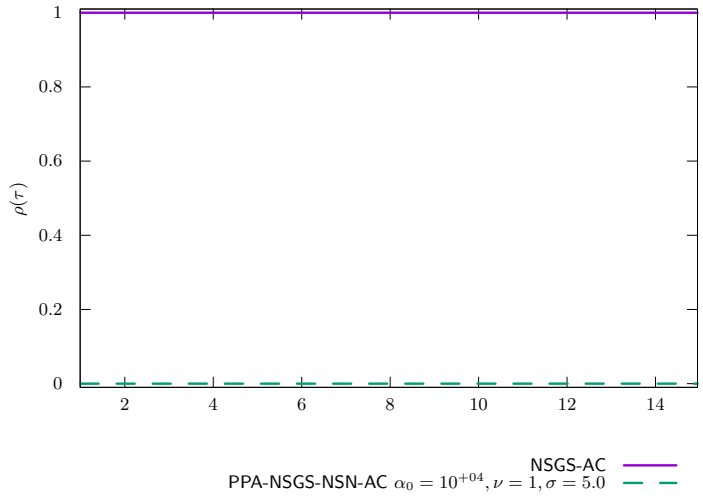


Figure 40: LMGC_100_PR_PerioBox time PROX/NSGS/InternalSolvers

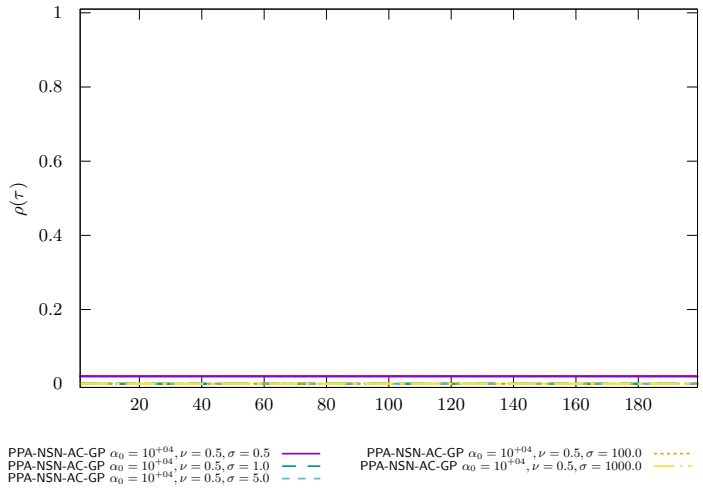


Figure 41: LMGC_100_PR_PerioBox time PROX/Parametric studies $\nu = 0.5$

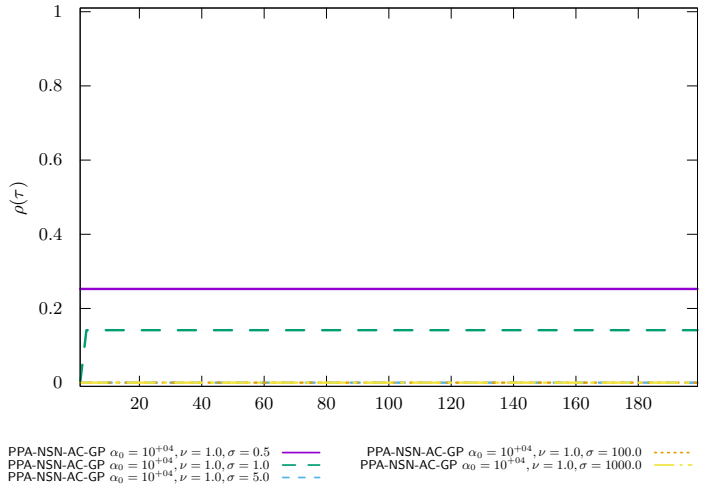


Figure 42: LMGC_100_PR_PerioBox time PROX/Parametric studies $\nu = 1.0$

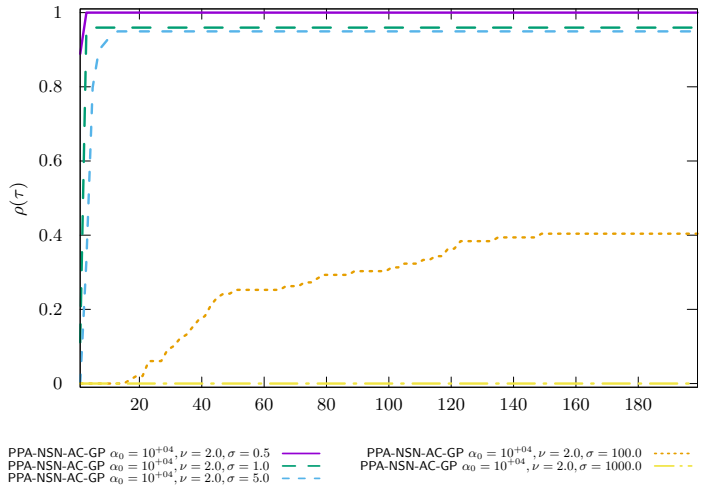


Figure 43: LMGC_100_PR_PerioBox time PROX/Parametric studies $\nu = 2.0$

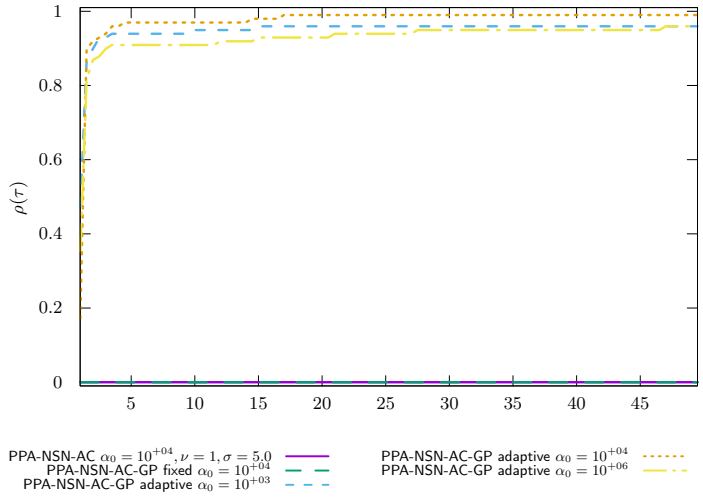


Figure 44: LMGC_100_PR_PerioBox time PROX/Regularized problem

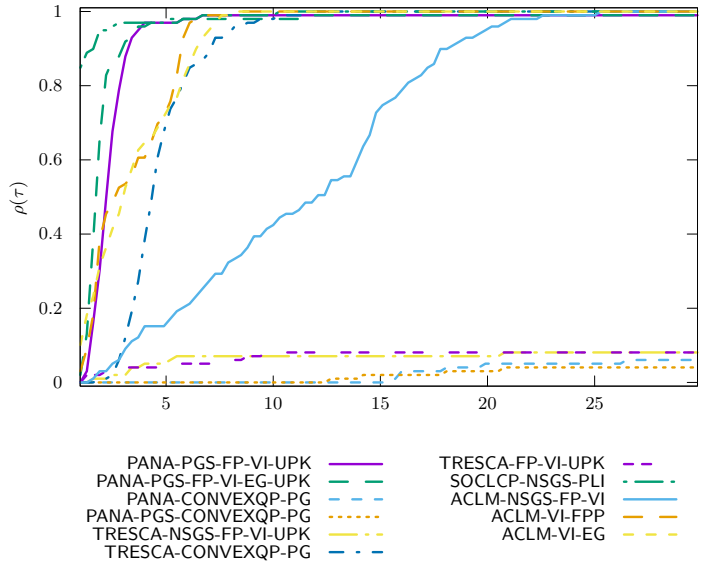


Figure 45: LMGC_100_PR_PerioBox time OPTI

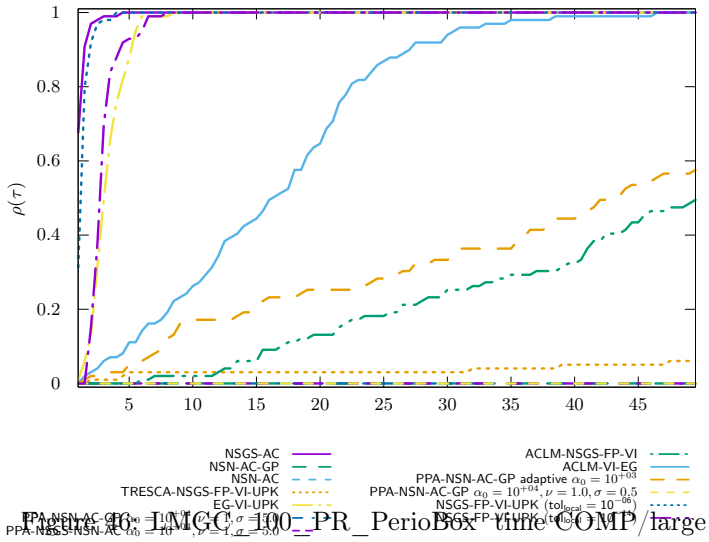


Figure 46: LMGC_100_PR_PerioBox time COMP/large

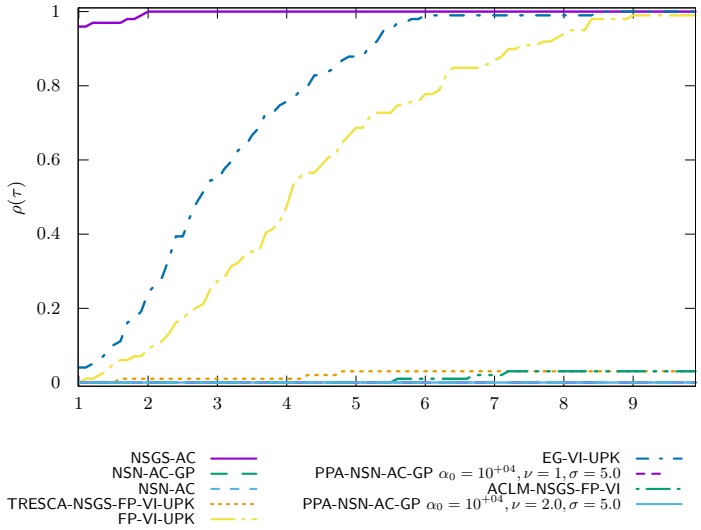


Figure 47: LMGC_100_PR_PerioBox time COMP/zoom

LMGC_945_SP_Box_PL precision 1.0e-04 timeout 100

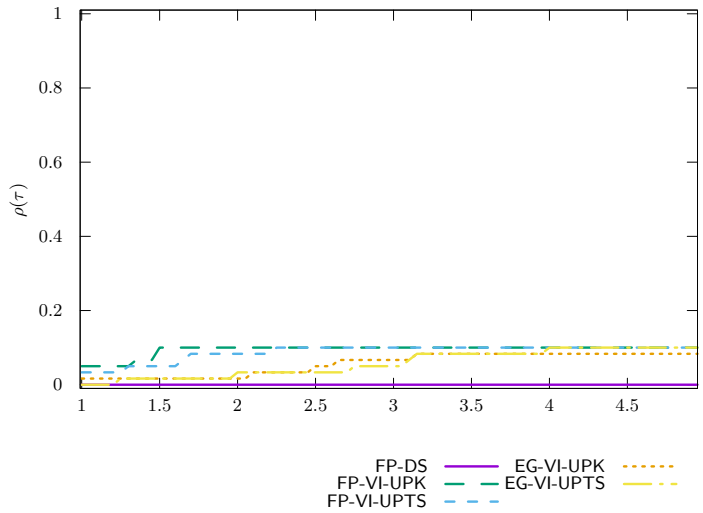


Figure 48: LMGC_945_SP_Box_PL time VI/UpdateRule

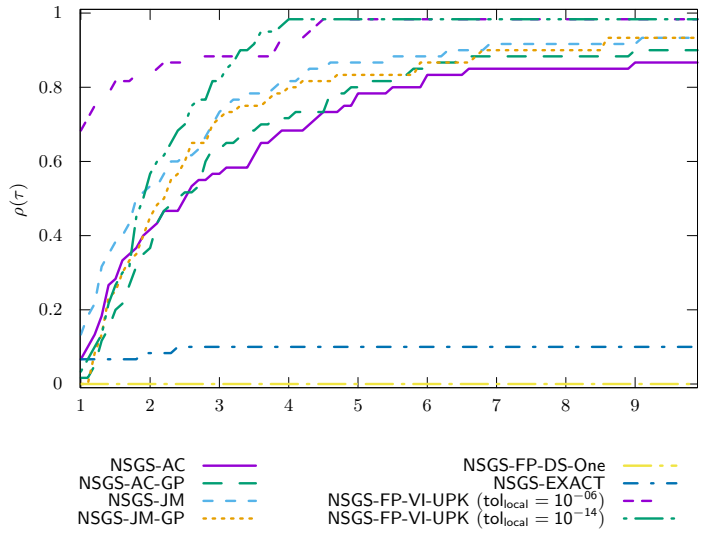


Figure 49: LMGC_945_SP_Box_PL time NSGS/LocalSolver

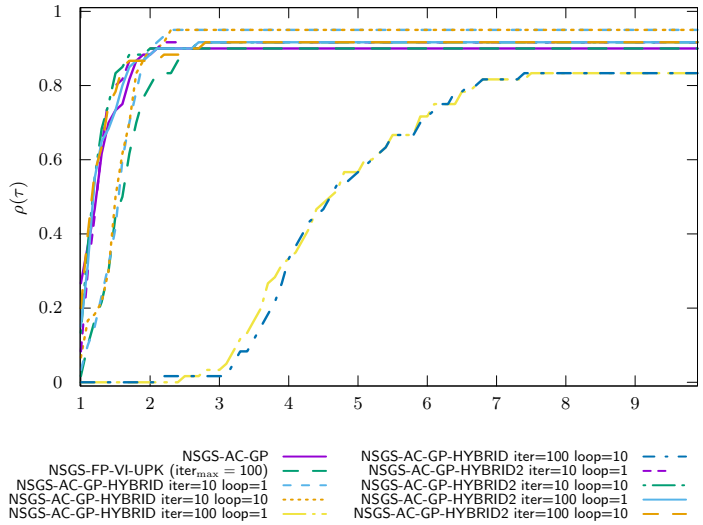


Figure 50: LMGC_945_SP_Box_PL time NSGS/LocalSolverHybrid

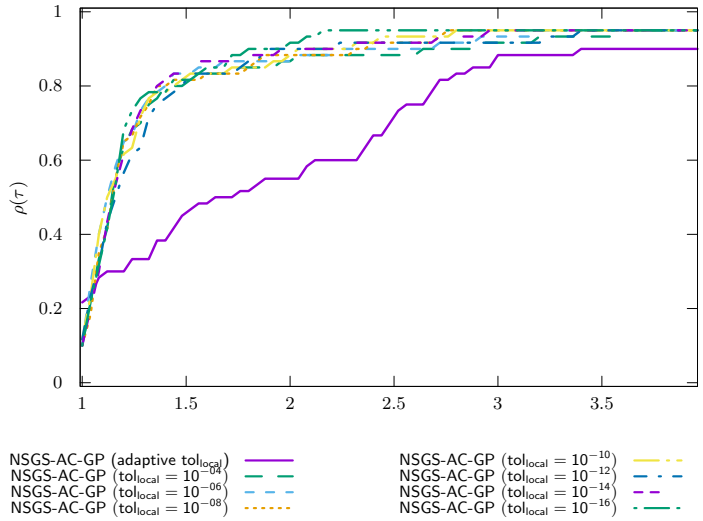


Figure 51: LMGC_945_SP_Box_PL time NSGS/LocalTol

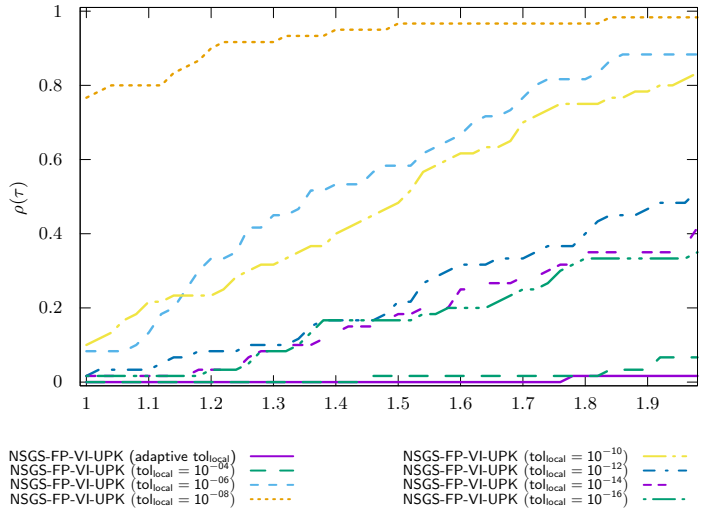


Figure 52: LMGC_945_SP_Box_PL time NSGS/LocalTol-VI

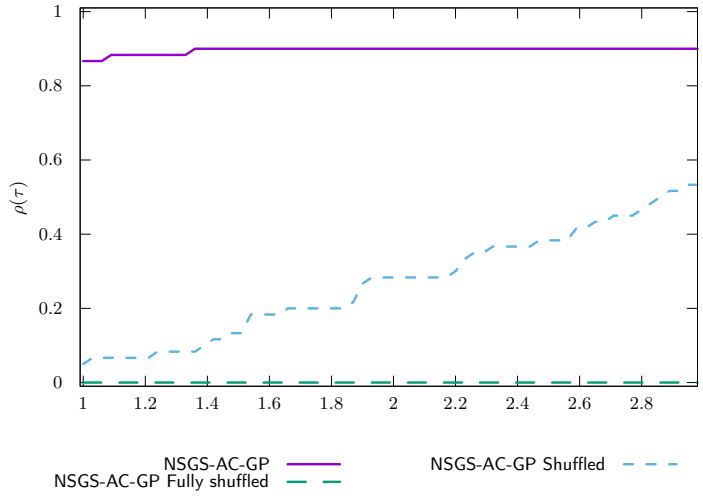


Figure 53: LMGC_945_SP_Box_PL time NSGS/Shuffled

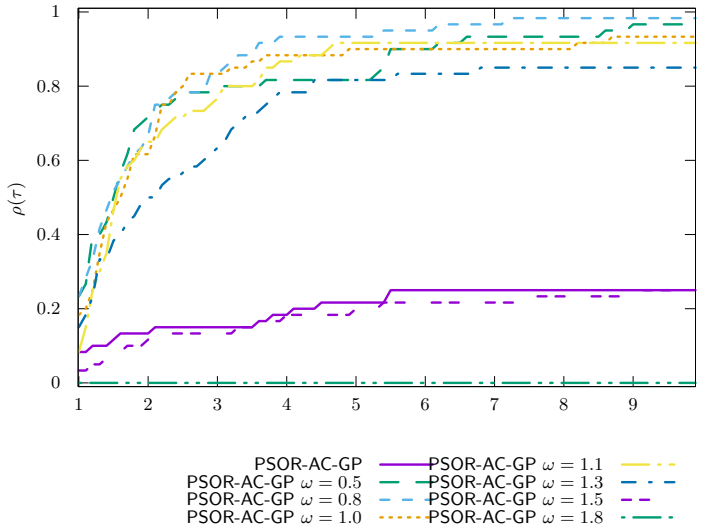


Figure 54: LMGC_945_SP_Box_PL time PSOR

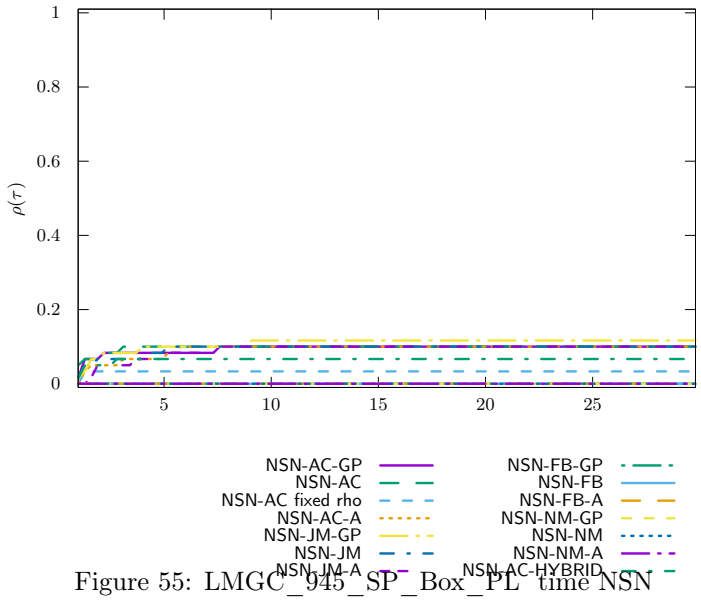


Figure 55: LMGC_945_SP_Box_PL time NSN

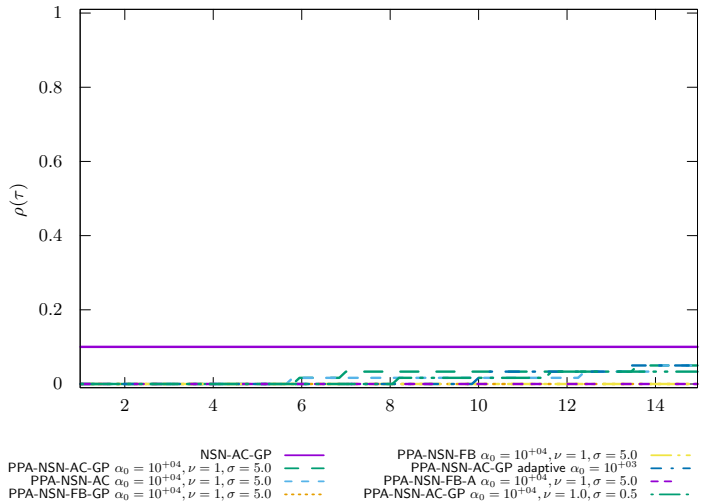


Figure 56: LMGC_945_SP_Box_PL time PROX/NSN/InternalSolvers

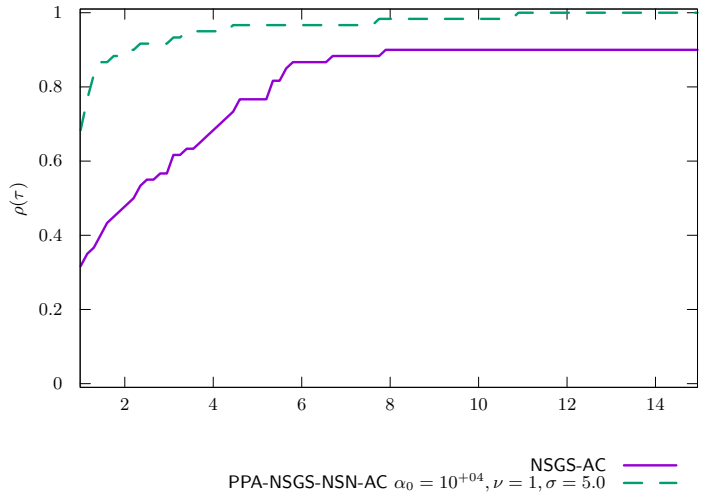


Figure 57: LMGC_945_SP_Box_PL time PROX/NSGS/InternalSolvers

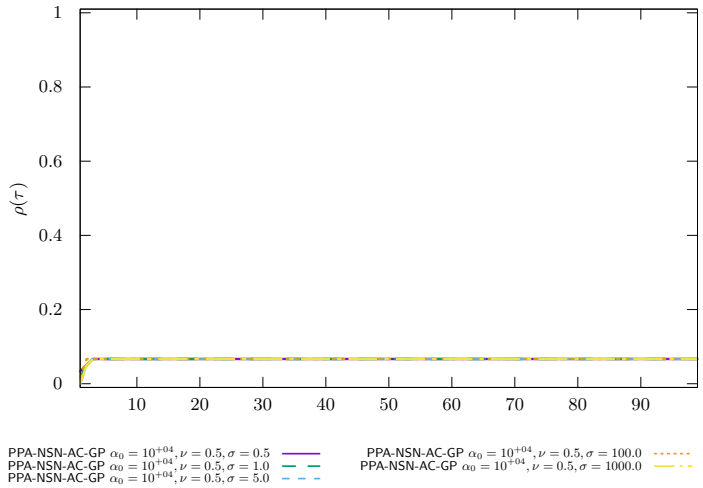


Figure 58: LMGC_945_SP_Box_PL time PROX/Parametric studies $\nu = 0.5$

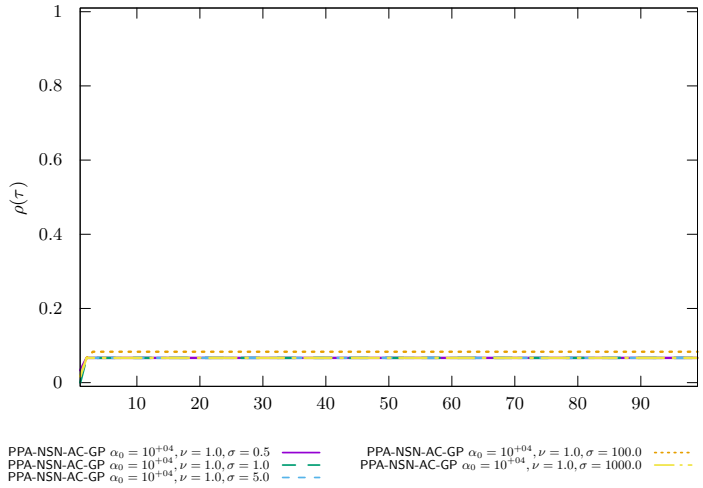


Figure 59: LMGC_945_SP_Box_PL time PROX/Parametric studies $\nu = 1.0$

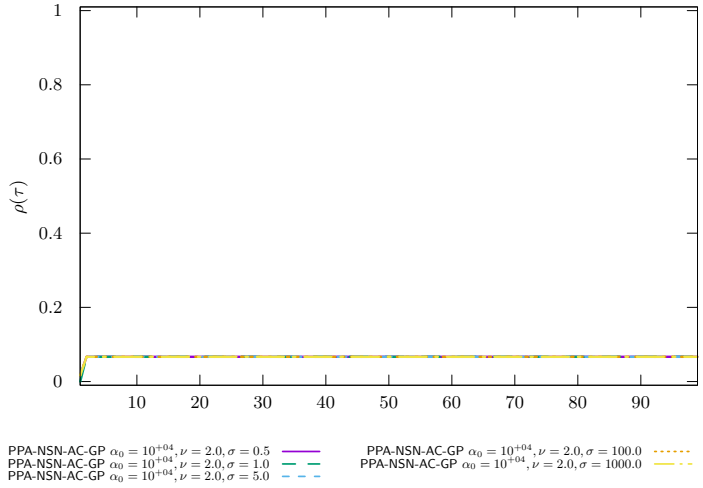


Figure 60: LMGC_945_SP_Box_PL time PROX/Parametric studies $\nu = 2.0$

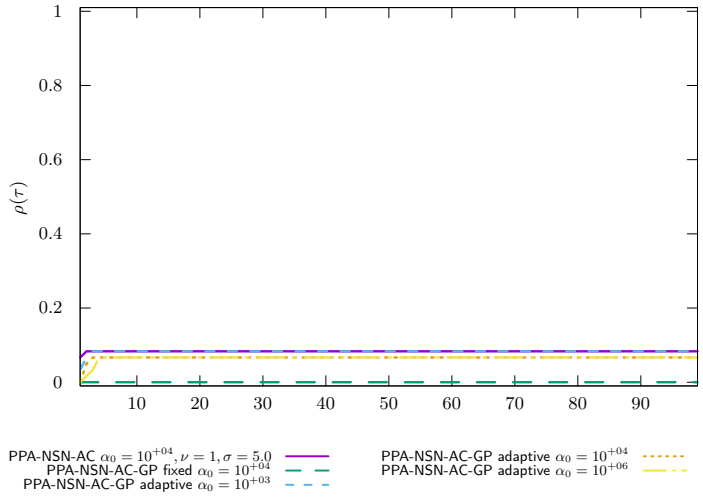


Figure 61: LMGC_945_SP_Box_PL time PROX/Regularized problem

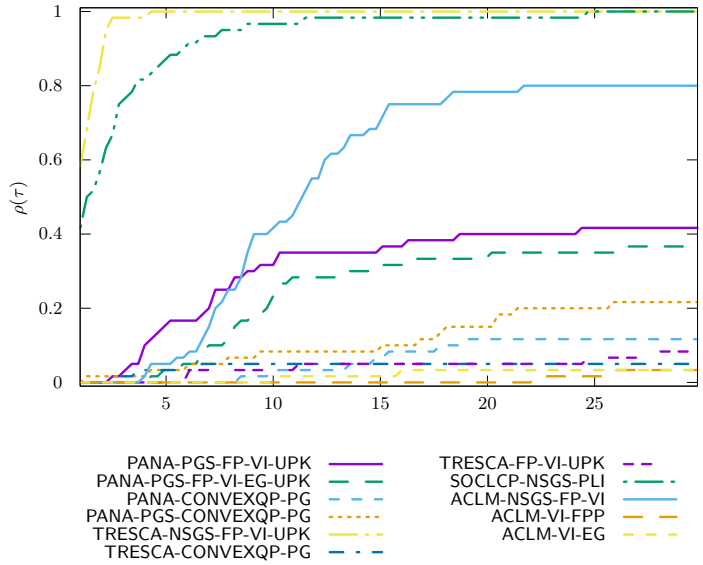


Figure 62: LMGC_945_SP_Box_PL time OPTI

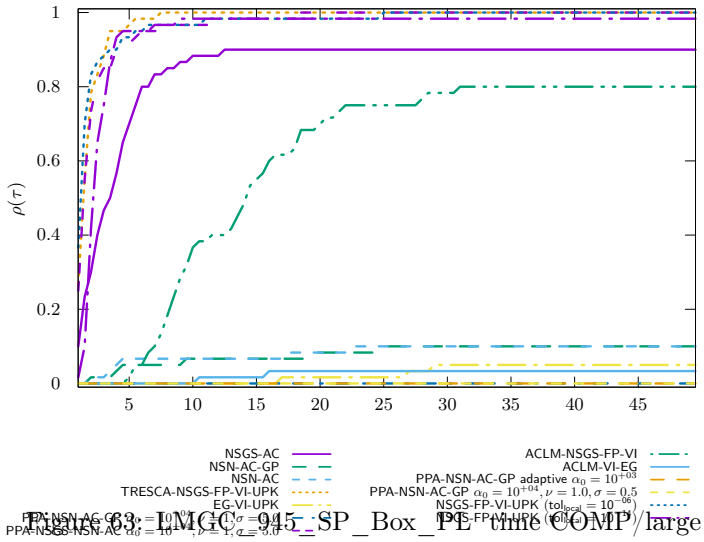


Figure 63: LMGC_945_SP_Box_PL time COMP/large

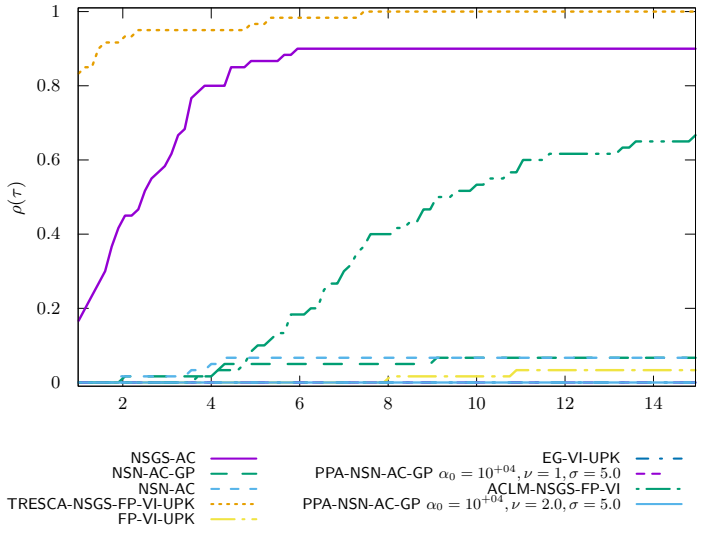


Figure 64: LMGC_945_SP_Box_PL time COMP/zoom

LMGC Aqueduc PR precision 1.0e-04 timeout 200

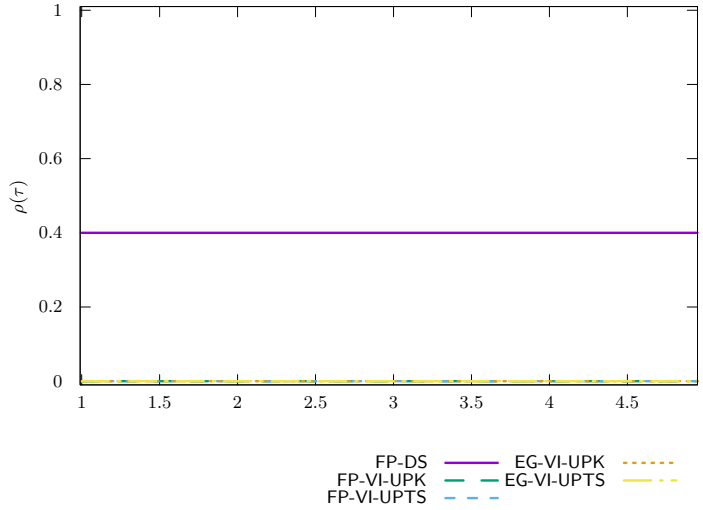


Figure 65: LMGC Aqueduc PR time VI/UpdateRule

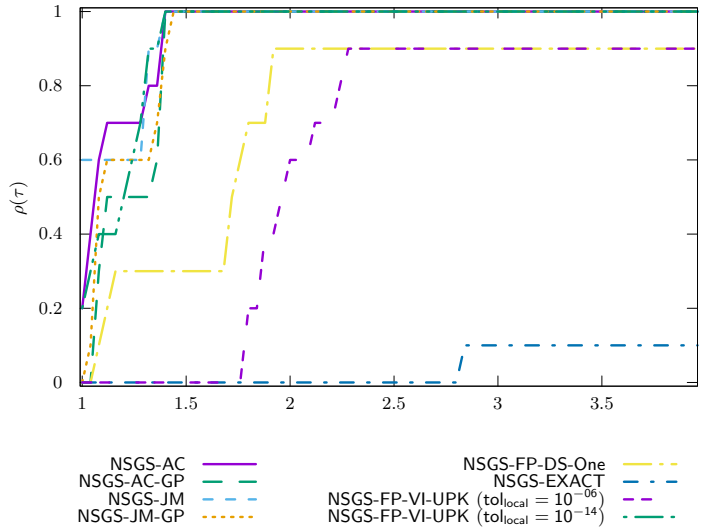


Figure 66: LMGC Aqueduc PR time NSGS/LocalSolver

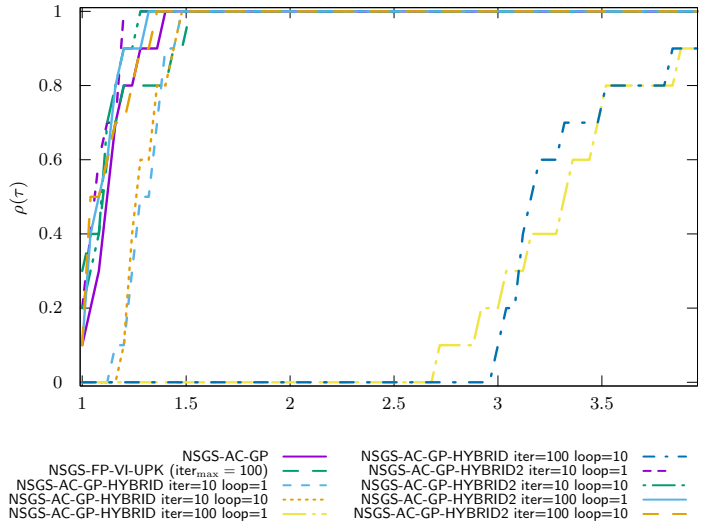


Figure 67: LMGC Aqueduc PR time NSGS/LocalSolverHybrid

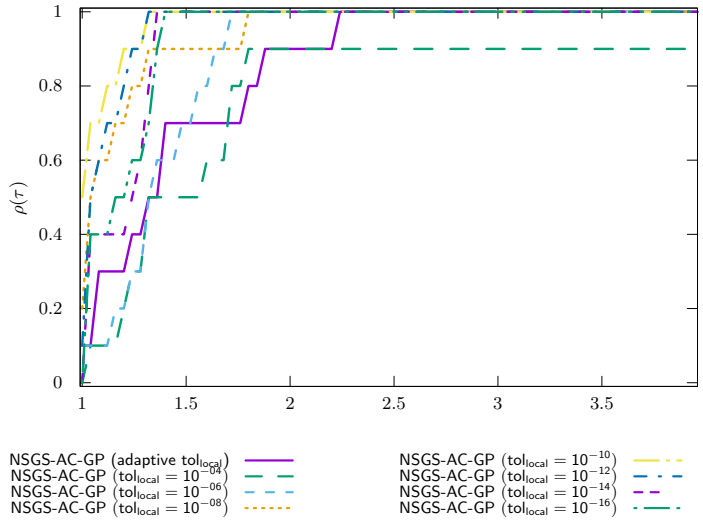


Figure 68: LMGC Aqueduc PR time NSGS/LocalTol

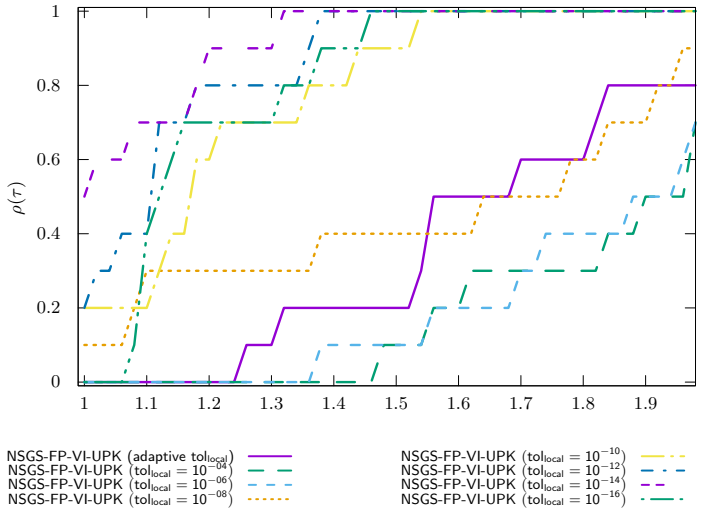


Figure 69: LMGC Aqueduc PR time NSGS/LocalTol-VI

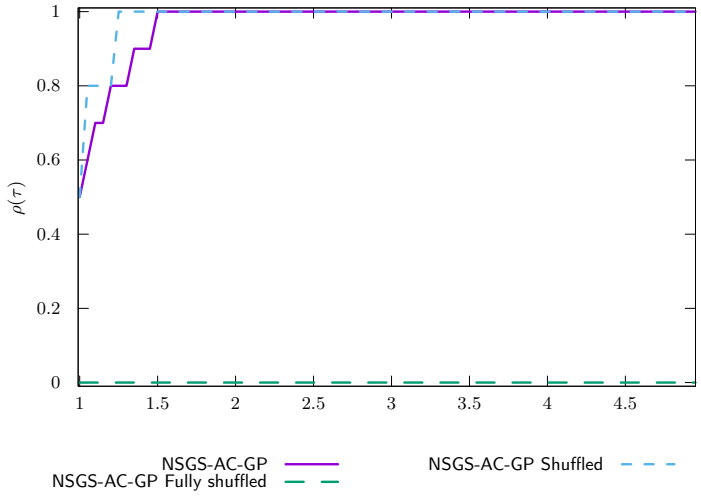


Figure 70: LMGC Aqueduc PR time NSGS/Shuffled

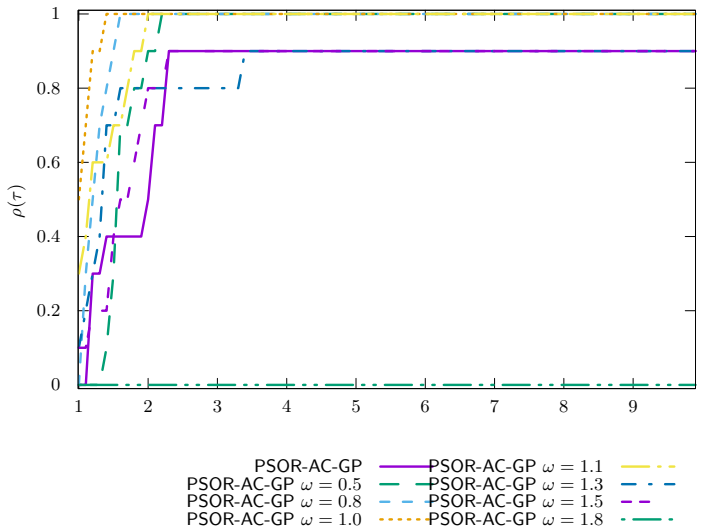


Figure 71: LMGC Aqueduc PR time PSOR

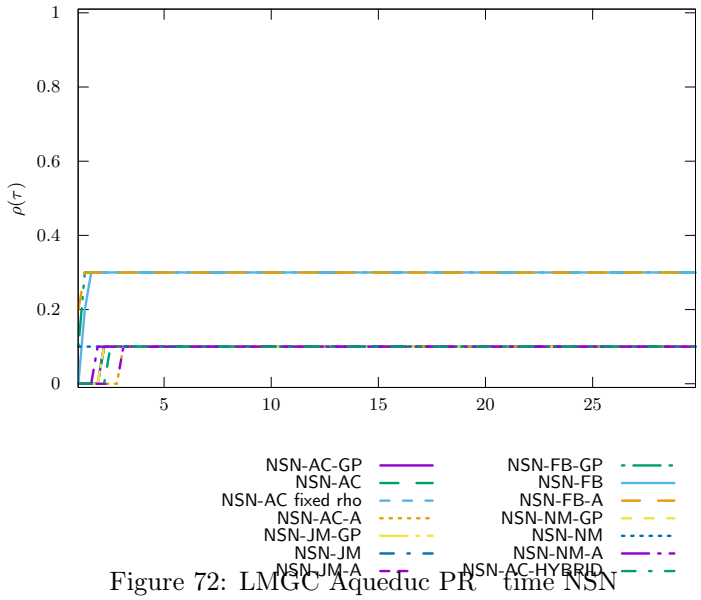


Figure 72: LMGC Aqueduc PR time NSN

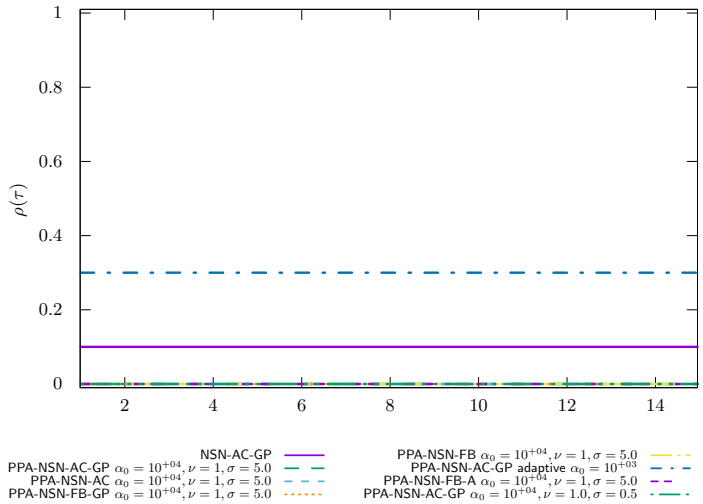


Figure 73: LMGC Aqueduc PR time PROX/NSN/InternalSolvers

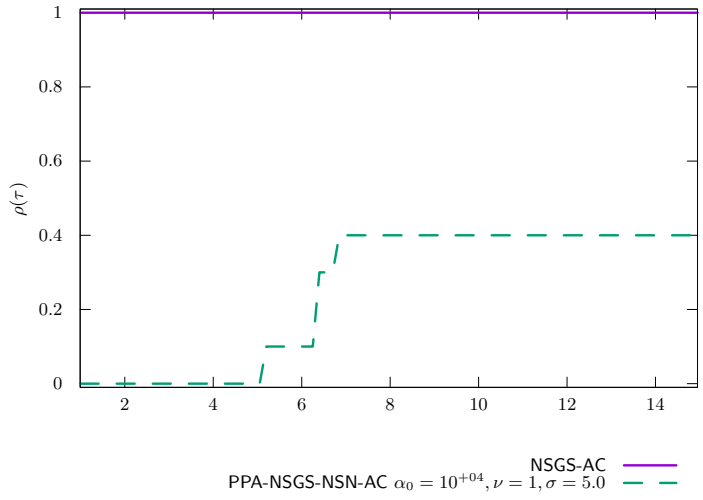


Figure 74: LMGC Aqueduc PR time PROX/NSGS/InternalSolvers

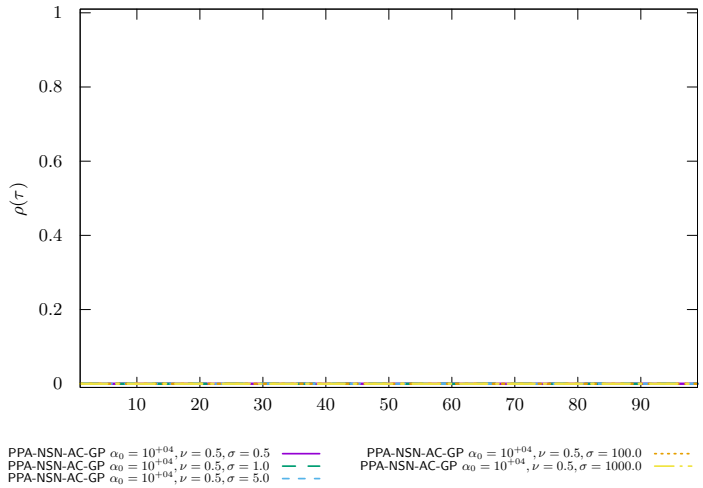


Figure 75: LMGC Aqueduc PR time PROX/Parametric studies $\nu = 0.5$

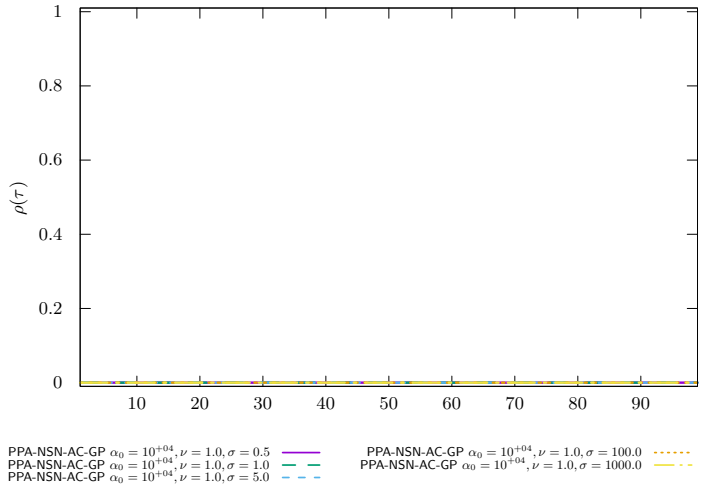


Figure 76: LMGC Aqueduc PR time PROX/Parametric studies $\nu = 1.0$

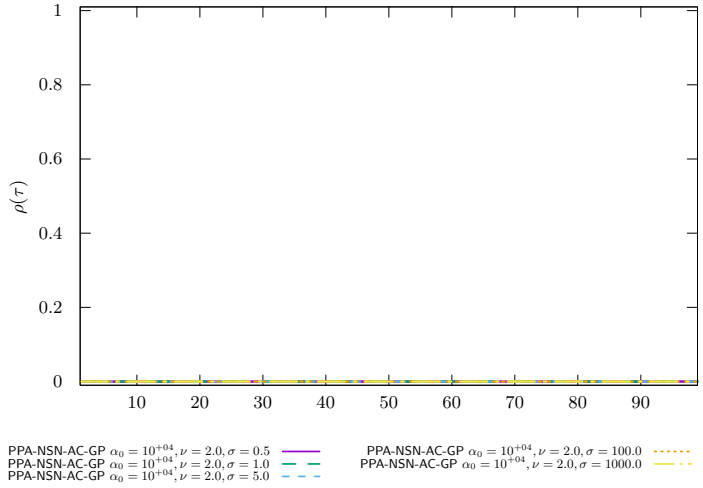


Figure 77: LMGC Aqueduc PR time PROX/Parametric studies $\nu = 2.0$

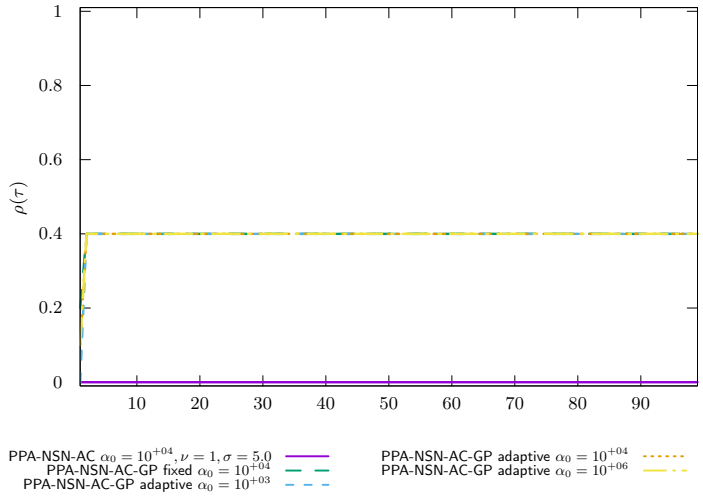


Figure 78: LMGC Aqueduc PR time PROX/Regularized problem

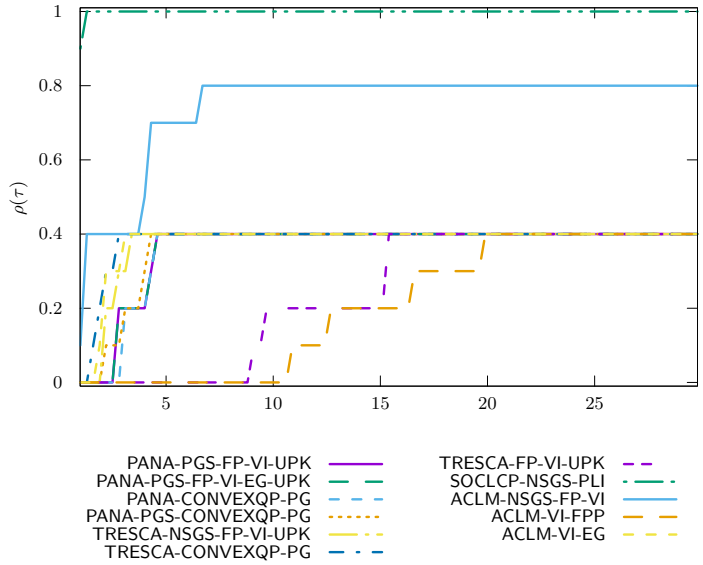


Figure 79: LMGC Aqueduc PR time OPTI

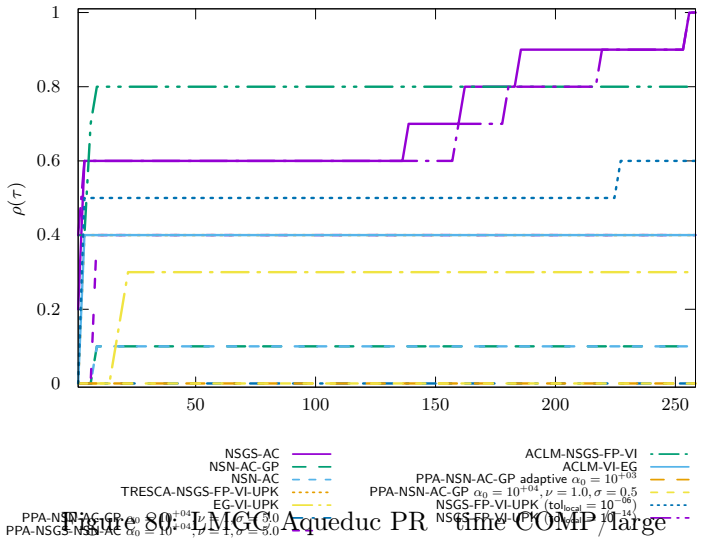


Figure 80: LMGC Aqueduc PR time COMP/large

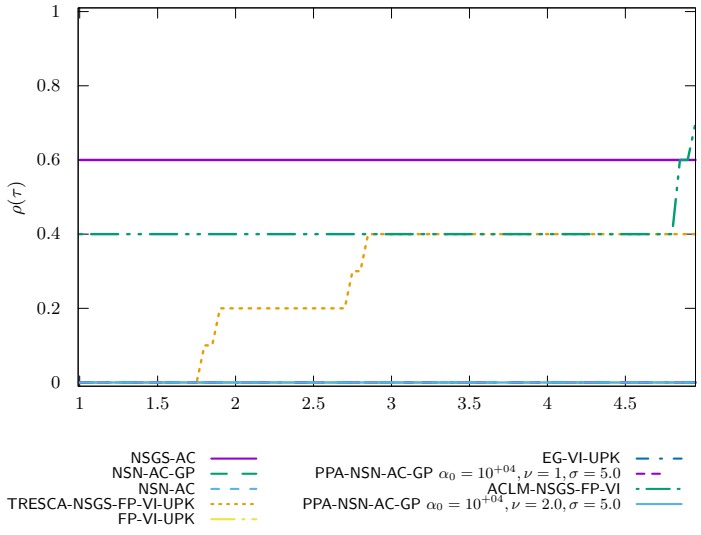


Figure 81: LMGC Aqueduc PR time COMP/zoom

LMGC Bridge PR precision 1.0e-04 timeout 100

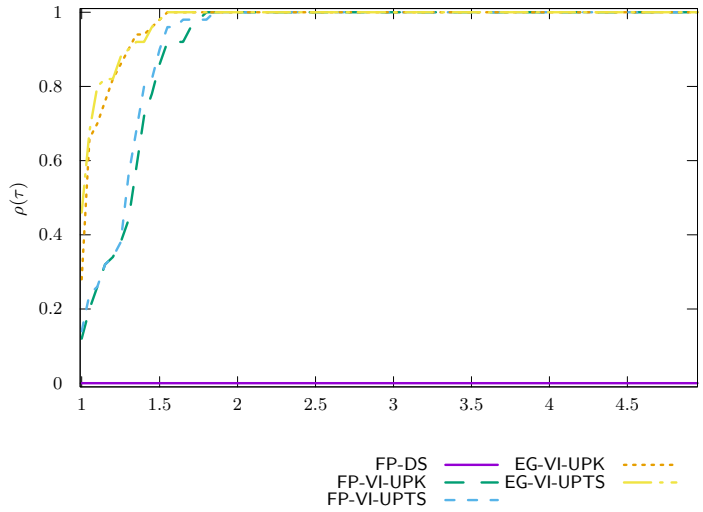


Figure 82: LMGC Bridge PR time VI/UpdateRule

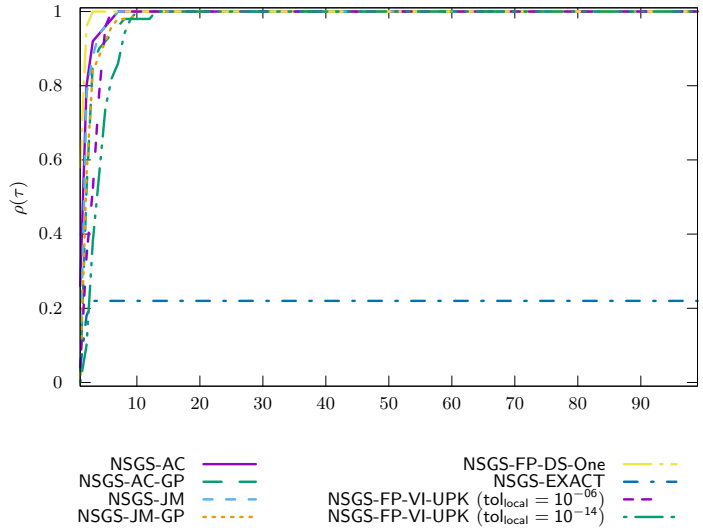


Figure 83: LMGC Bridge PR time NSGS/LocalSolver

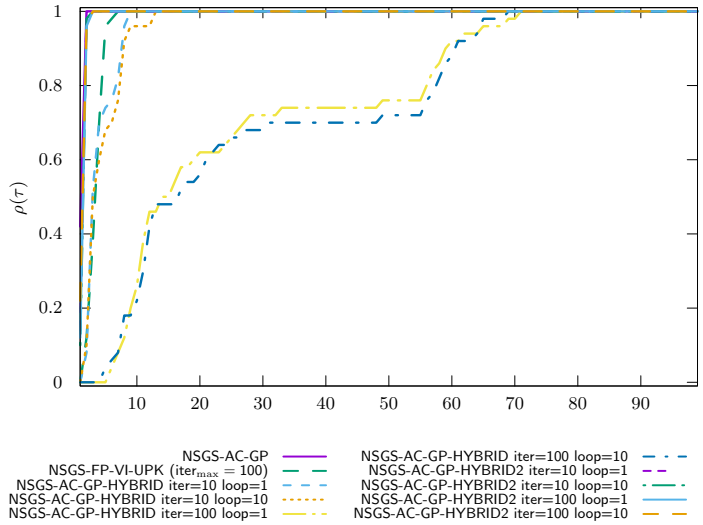


Figure 84: LMGC Bridge PR time NSGS/LocalSolverHybrid

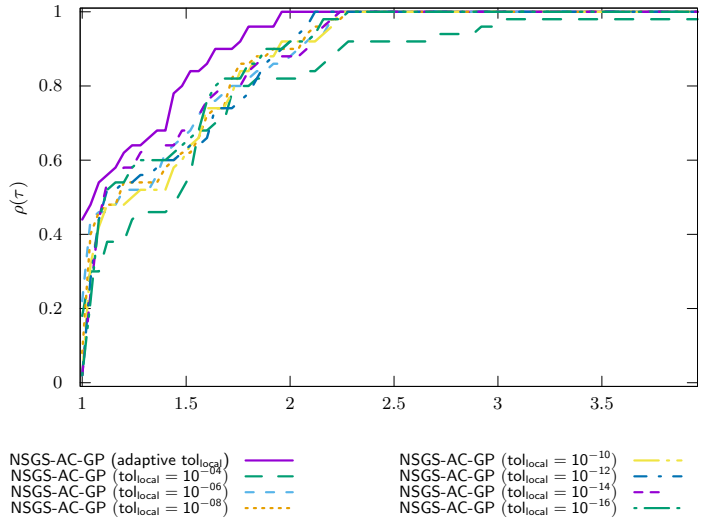


Figure 85: LMGC Bridge PR time NSGS/LocalTol

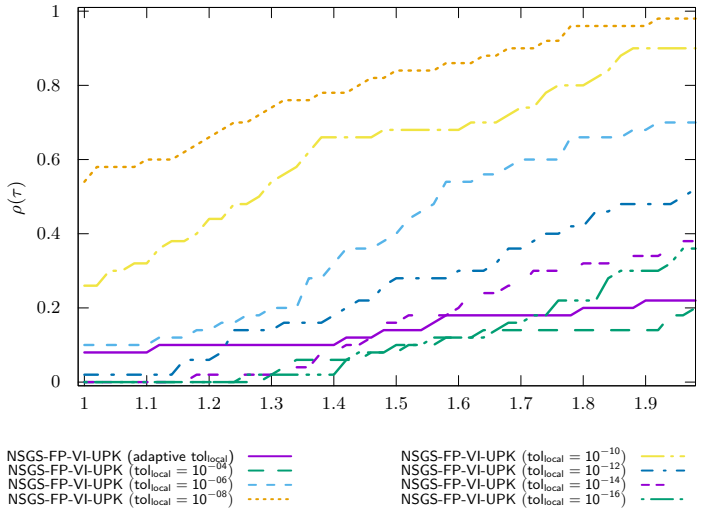


Figure 86: LMGC Bridge PR time NSGS/LocalTol-VI

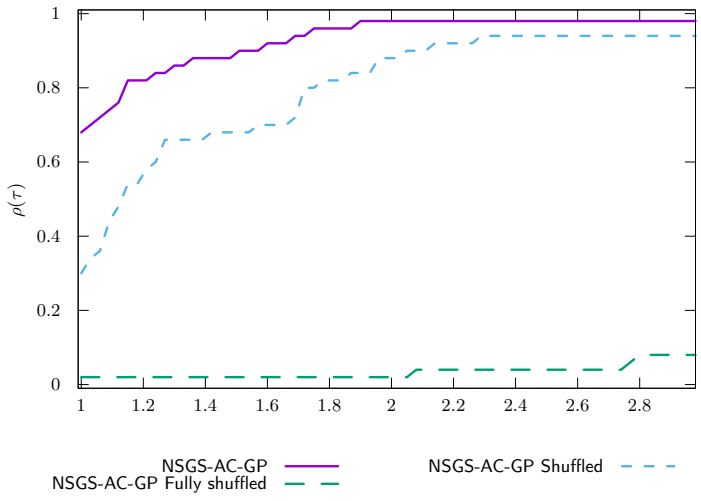


Figure 87: LMGC Bridge PR time NSGS/Shuffled

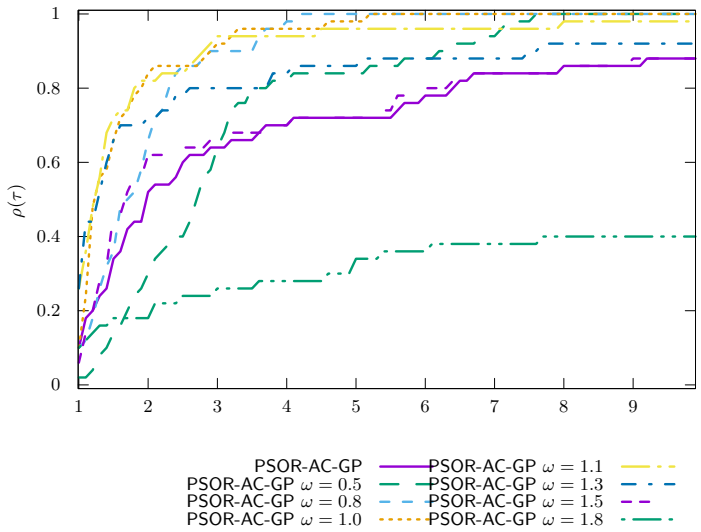


Figure 88: LMGC Bridge PR time PSOR

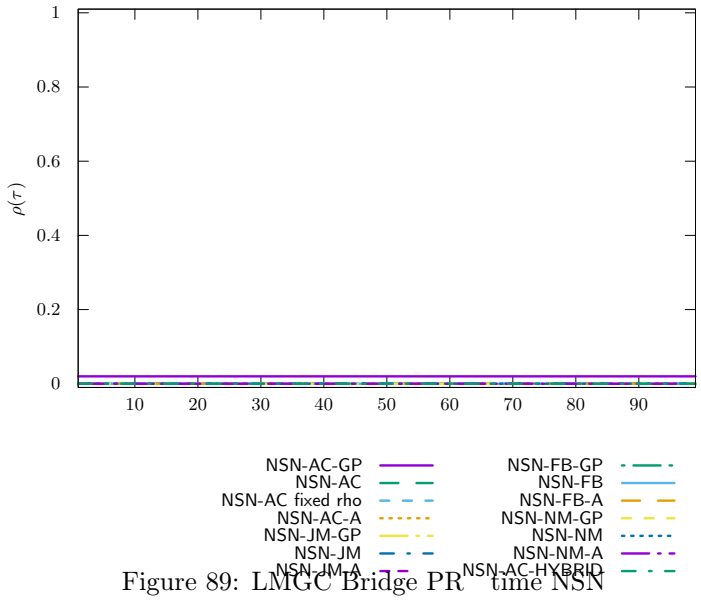


Figure 89: LMGC Bridge PR time NSN

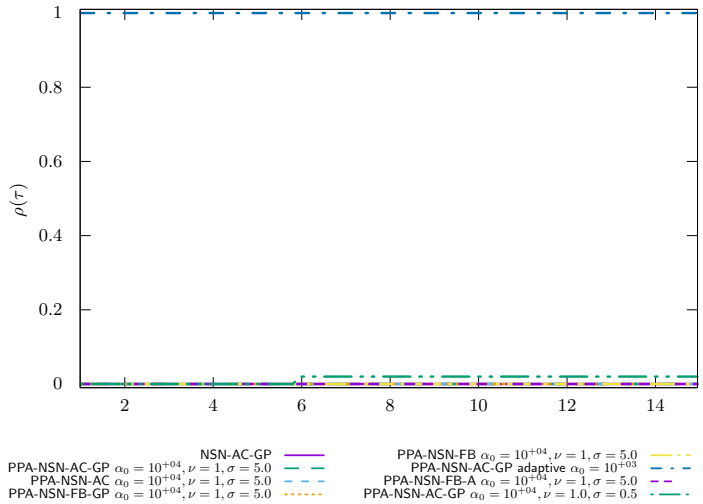


Figure 90: LMGC Bridge PR time PROX/NSN/InternalSolvers

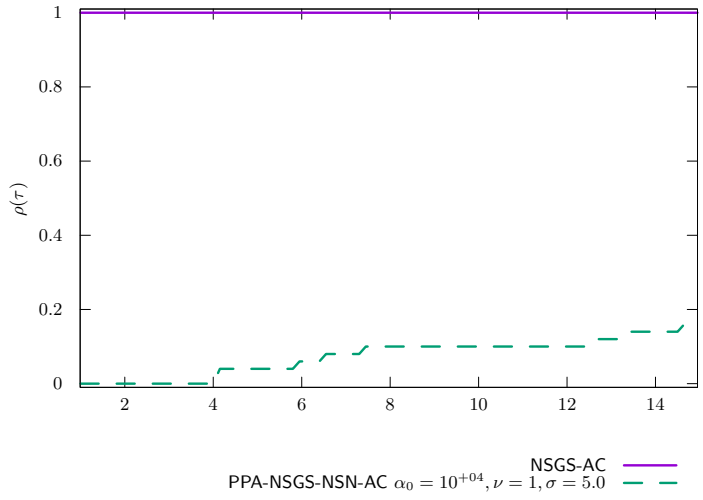


Figure 91: LMGC Bridge PR time PROX/NSGS/InternalSolvers

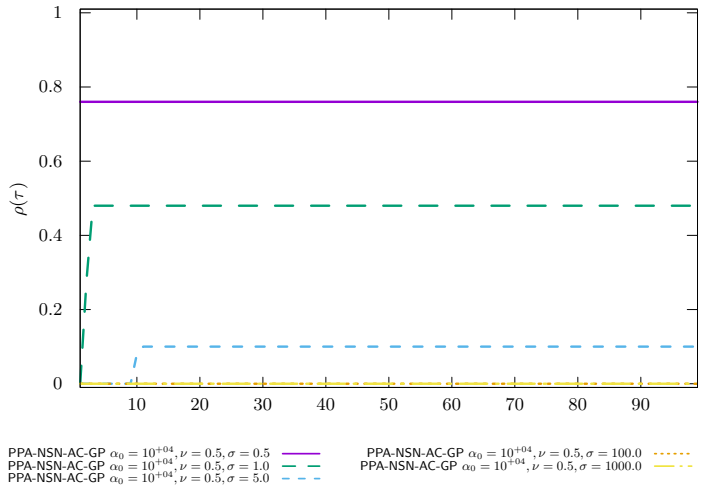


Figure 92: LMGC Bridge PR time PROX/Parametric studies $\nu = 0.5$

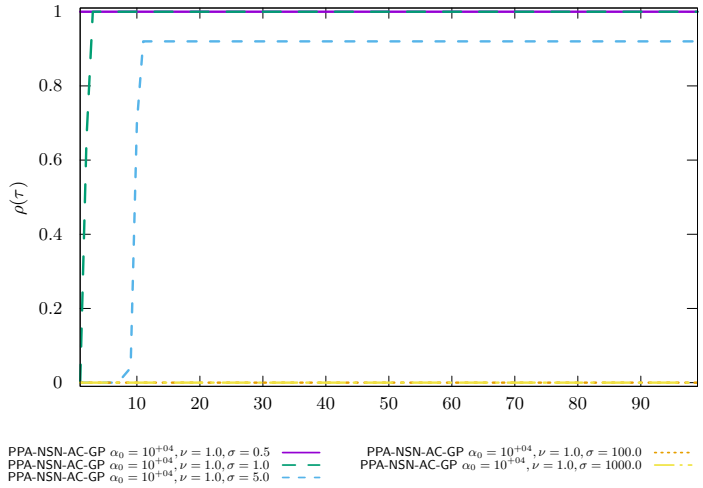


Figure 93: LMGC Bridge PR time PROX/Parametric studies $\nu = 1.0$

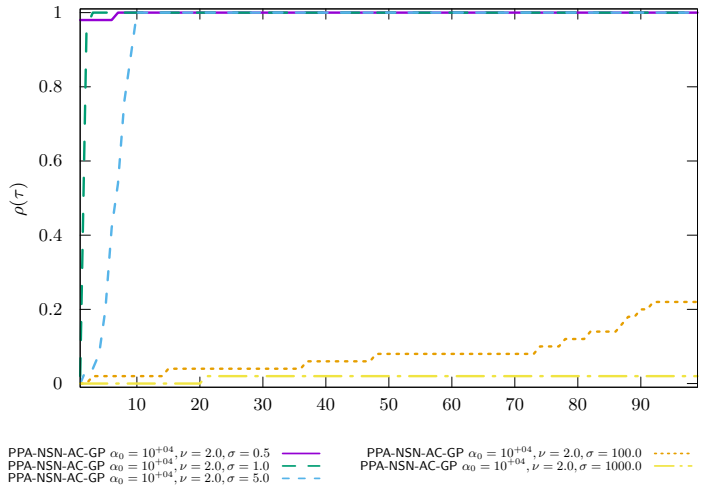


Figure 94: LMGC Bridge PR time PROX/Parametric studies $\nu = 2.0$

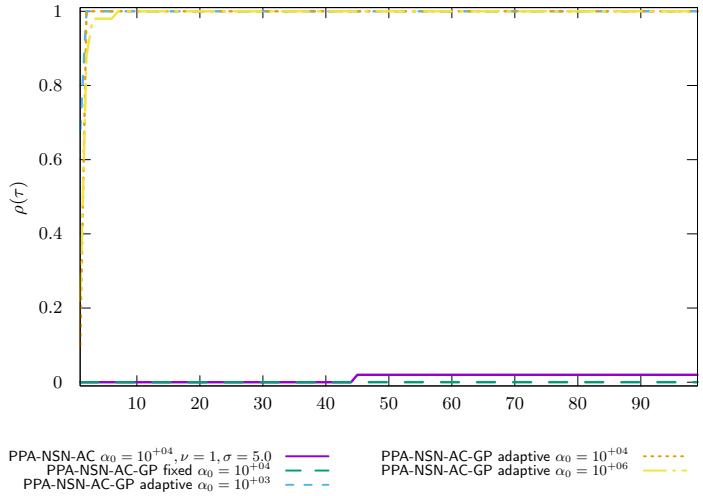


Figure 95: LMGC Bridge PR time PROX/Regularized problem

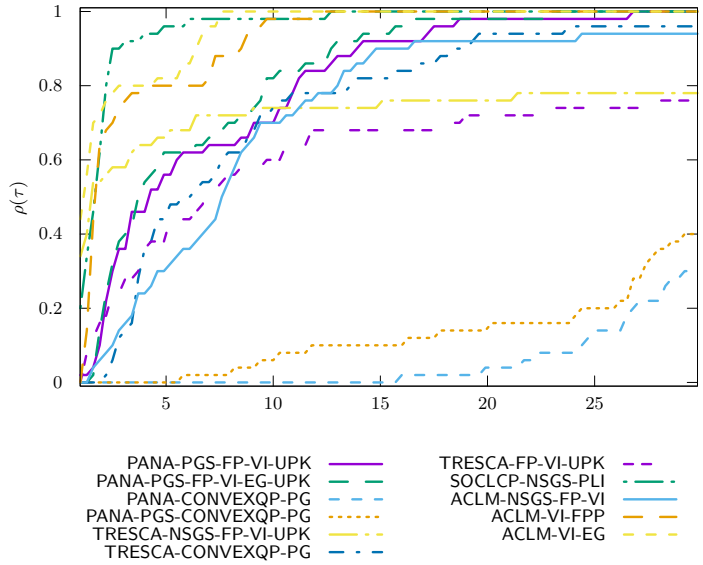


Figure 96: LMGC Bridge PR time OPTI

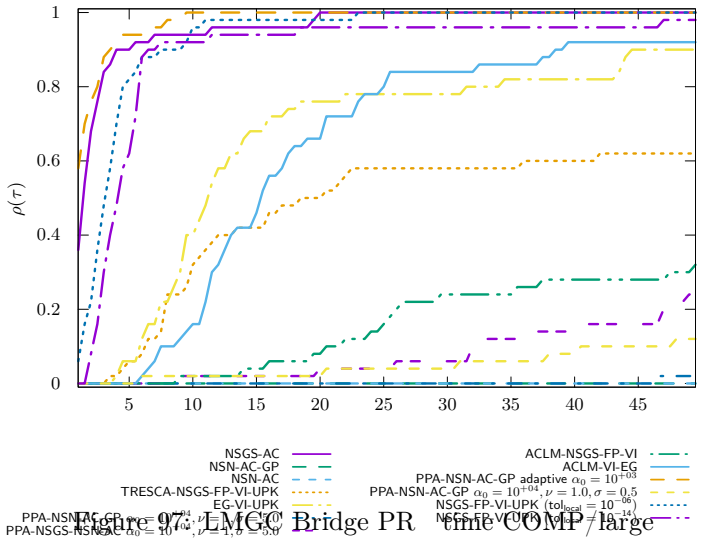


Figure 97: LMGC Bridge PR time COMP/large

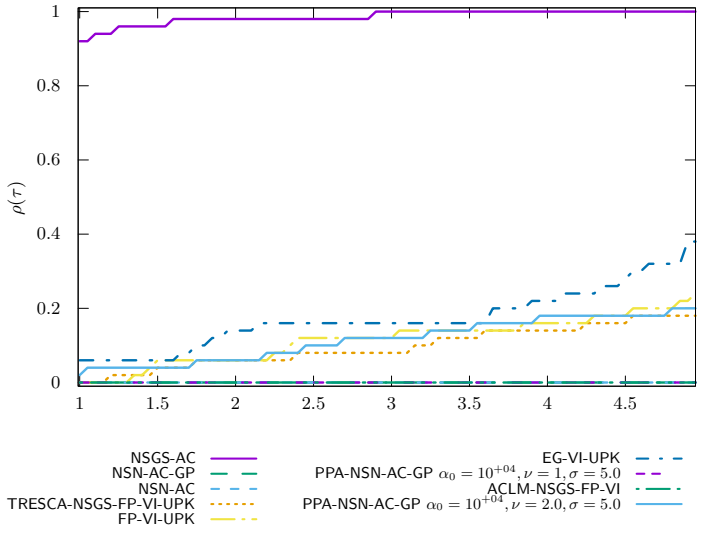


Figure 98: LMGC Bridge PR time COMP/zoom

LMGC Bridge PR precision 1.0e-08 timeout 400

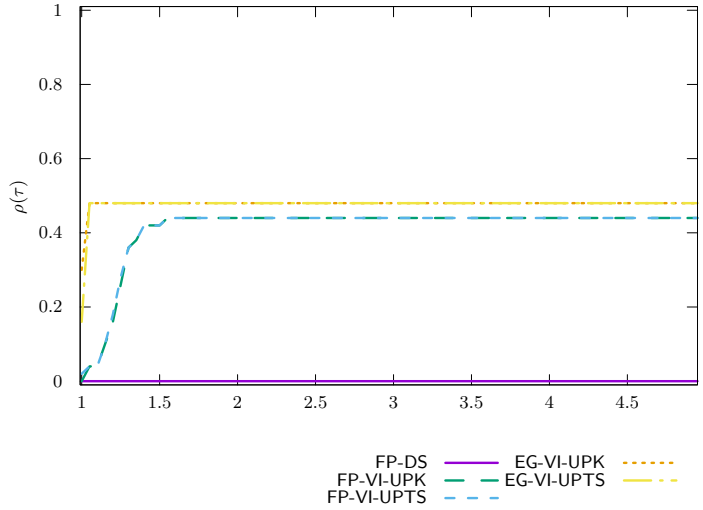


Figure 99: LMGC Bridge PR time VI/UpdateRule

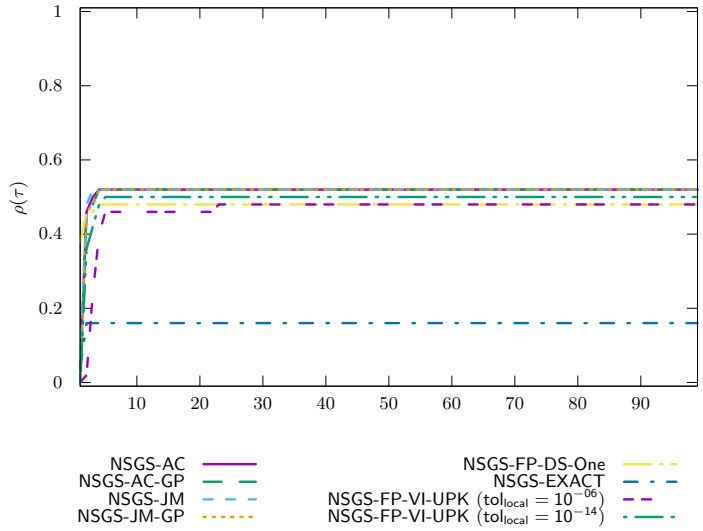


Figure 100: LMGC Bridge PR time NSGS/LocalSolver

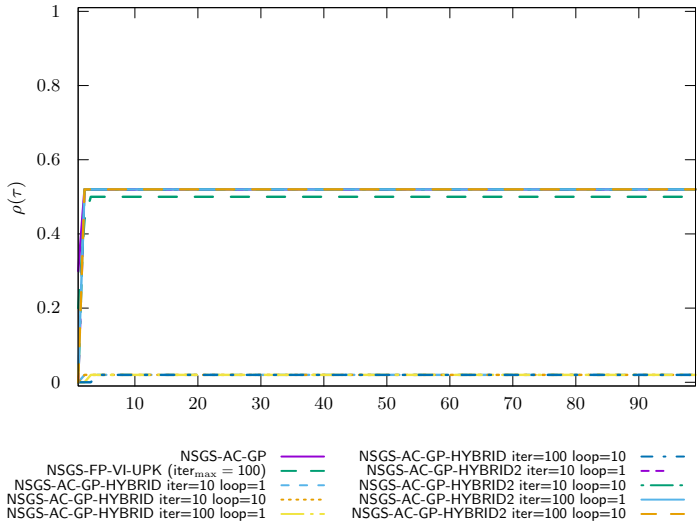


Figure 101: LMGC Bridge PR time NSGS/LocalSolverHybrid

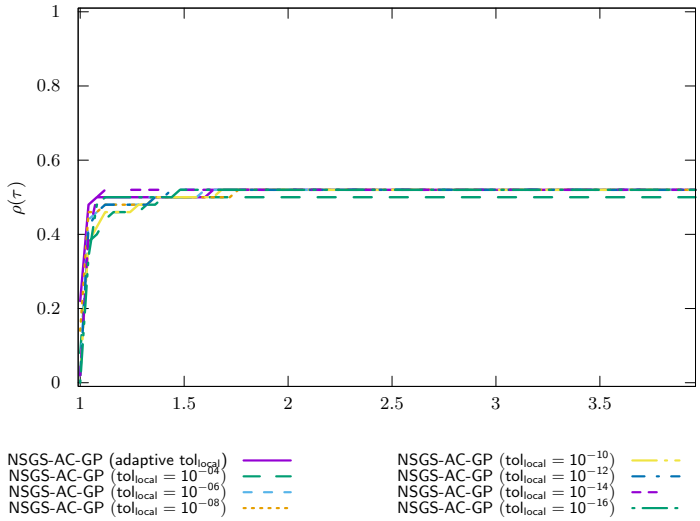


Figure 102: LMGC Bridge PR time NSGS/LocalTol

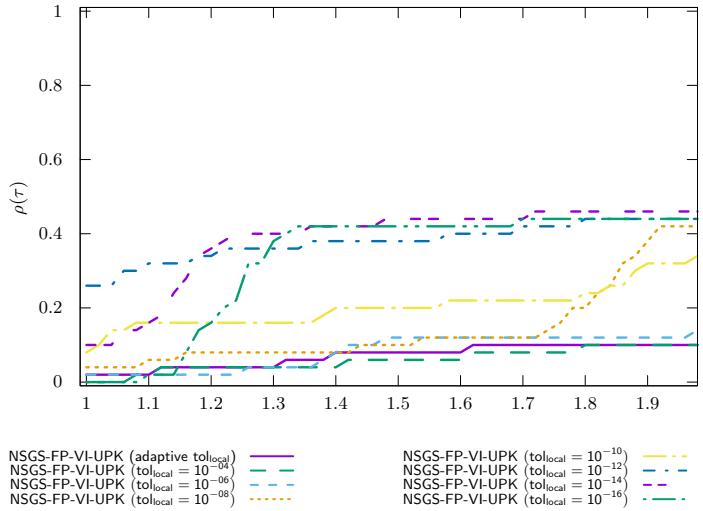


Figure 103: LMGC Bridge PR time NSGS/LocalTol-VI

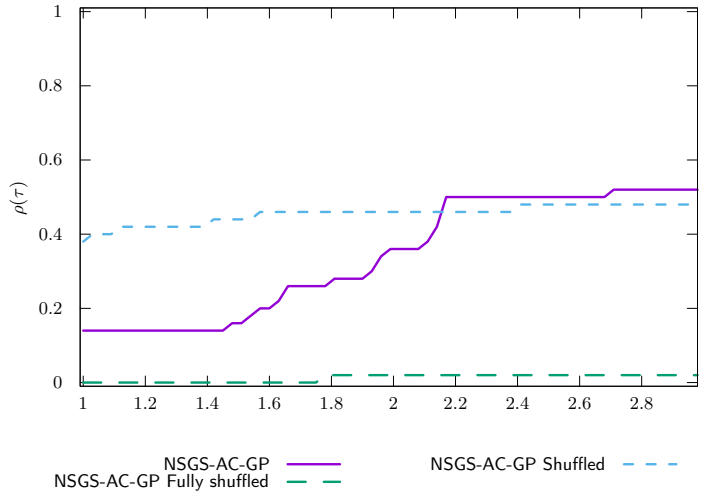


Figure 104: LMGC Bridge PR time NSGS/Shuffled

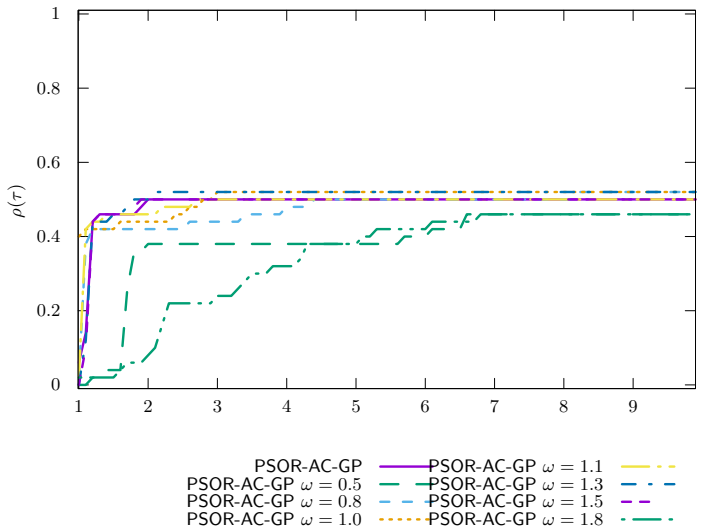


Figure 105: LMGC Bridge PR time PSOR

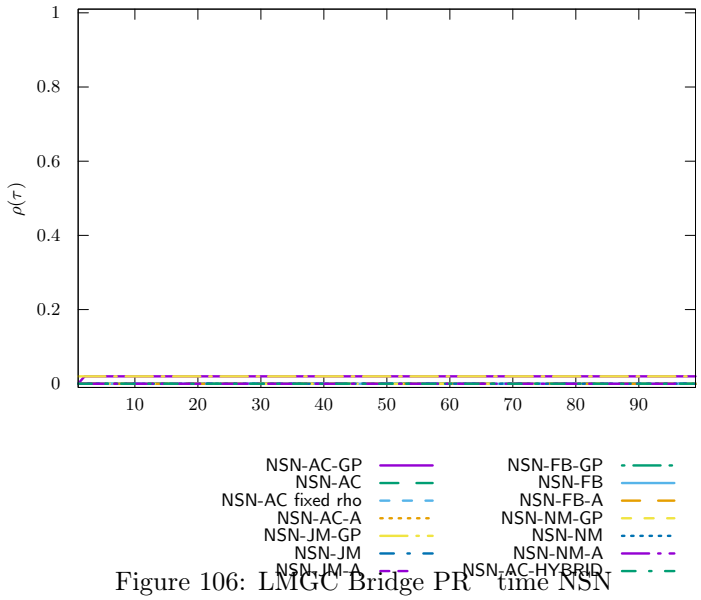


Figure 106: LMGC Bridge PR time NSN

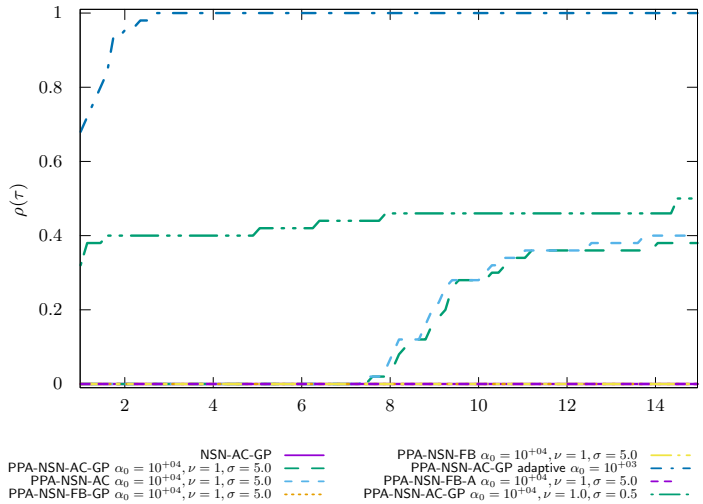


Figure 107: LMGC Bridge PR time PROX/NSN/InternalSolvers

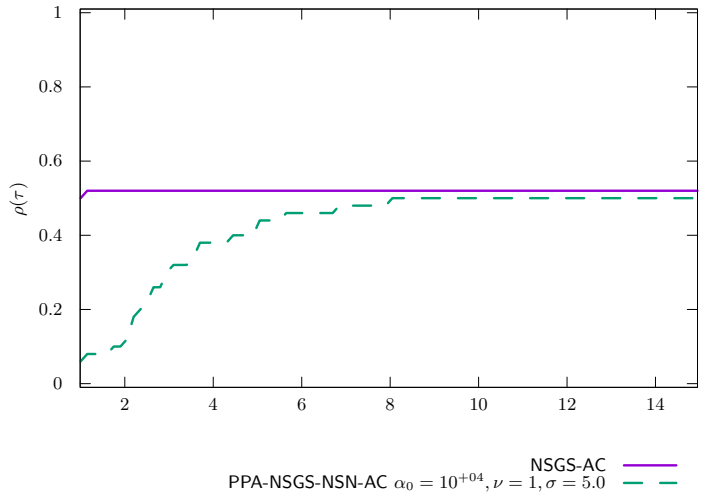


Figure 108: LMGC Bridge PR time PROX/NSGS/InternalSolvers

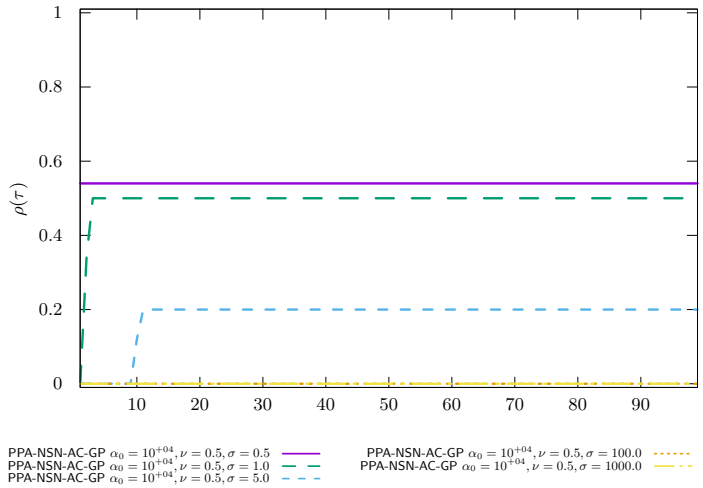


Figure 109: LMGC Bridge PR time PROX/Parametric studies $\nu = 0.5$

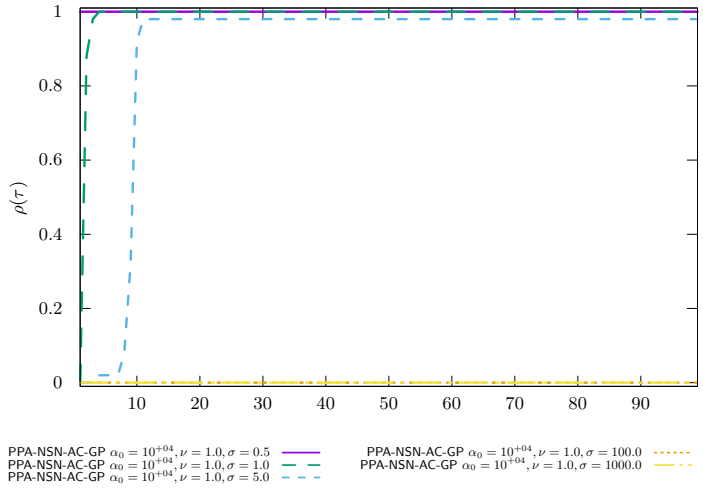


Figure 110: LMGC Bridge PR time PROX/Parametric studies $\nu = 1.0$

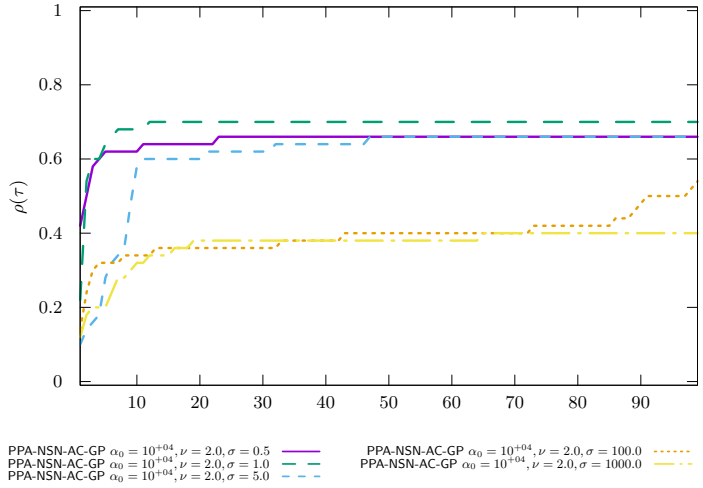


Figure 111: LMGC Bridge PR time PROX/Parametric studies $\nu = 2.0$

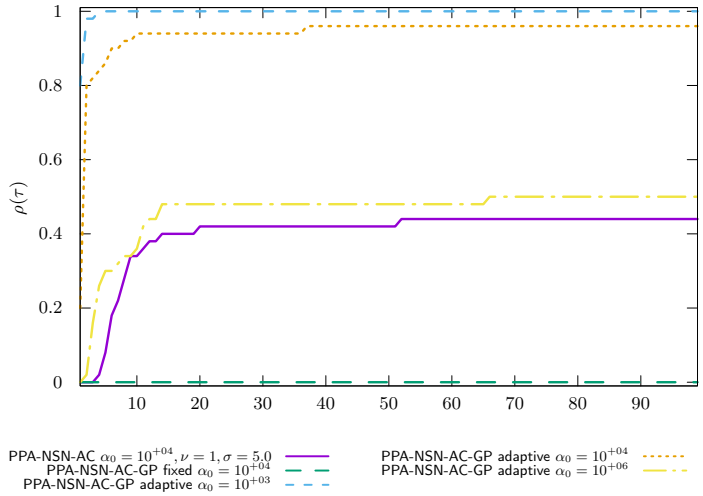


Figure 112: LMGC Bridge PR time PROX/Regularized problem

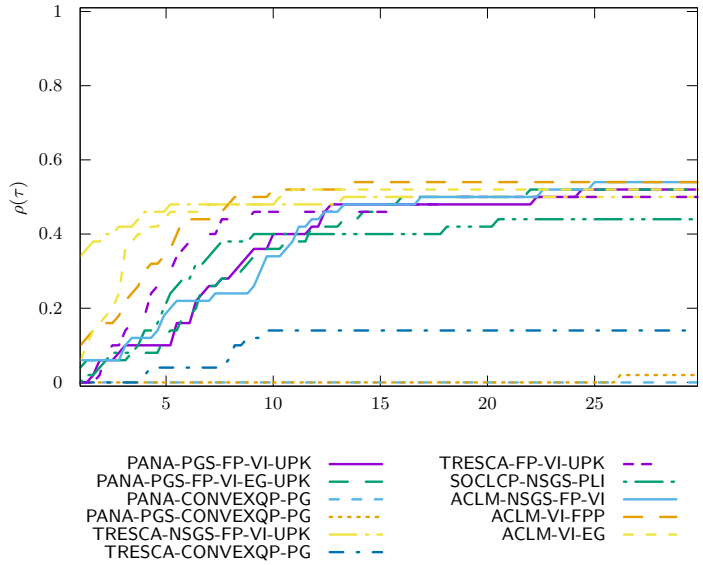


Figure 113: LMGC Bridge PR time OPTI

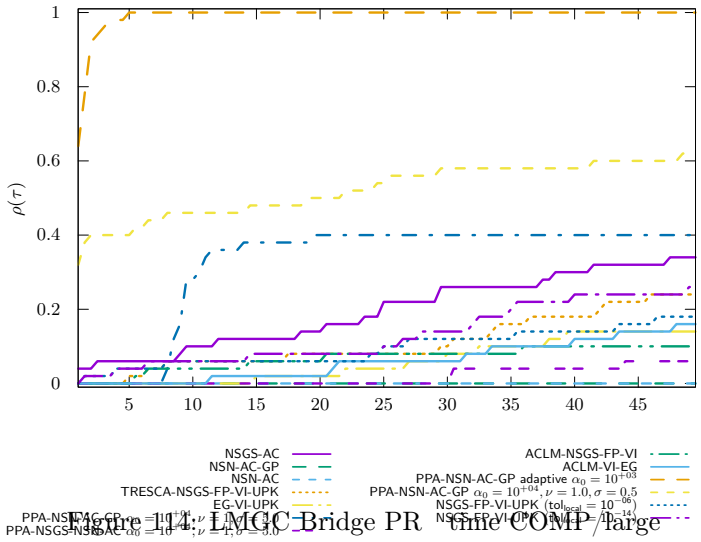


Figure 114: LMGC Bridge PR time COMP/large

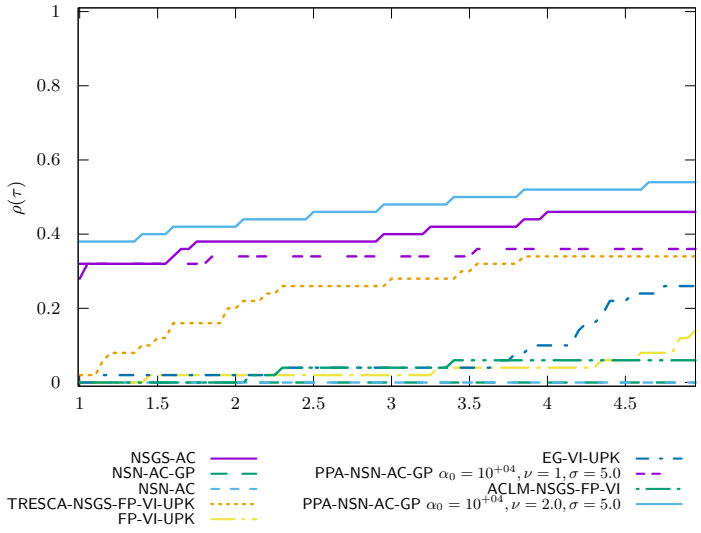


Figure 115: LMGC Bridge PR time COMP/zoom

LMGC LowWall FEM precision 1.0e-04 timeout 400

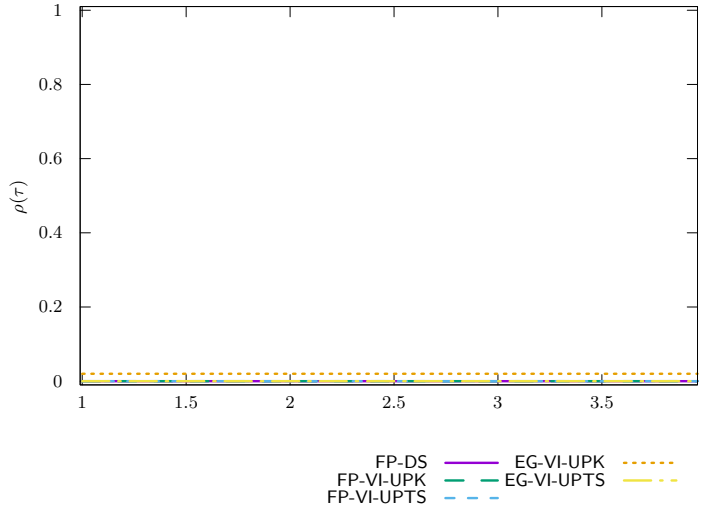


Figure 116: LMGC LowWall FEM time VI/UpdateRule

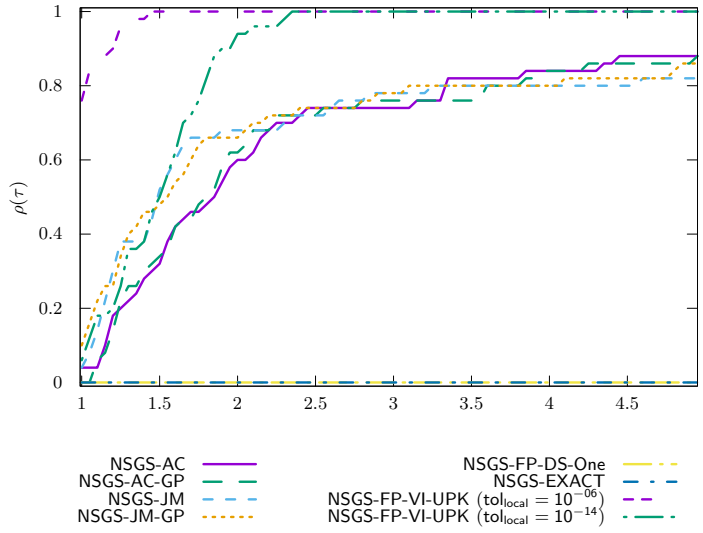


Figure 117: LMGC LowWall FEM time NSGS/LocalSolver

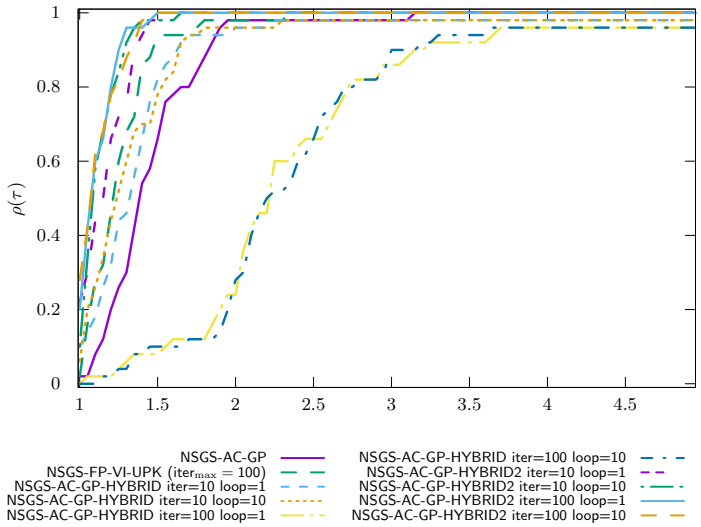


Figure 118: LMGC LowWall FEM time NSGS/LocalSolverHybrid

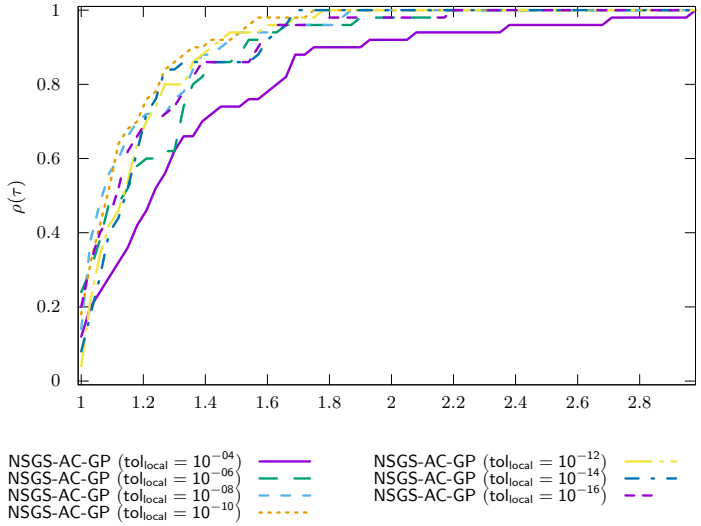


Figure 119: LMGC LowWall FEM time NSGS/LocalTol

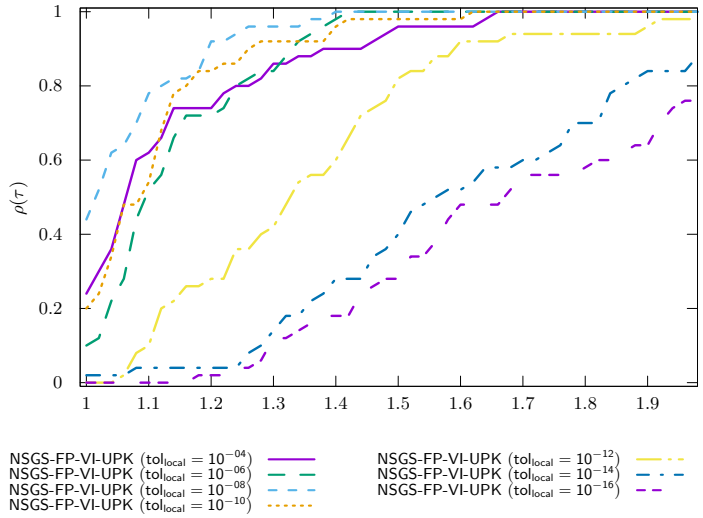


Figure 120: LMGC LowWall FEM time NSGS/LocalTol-VI

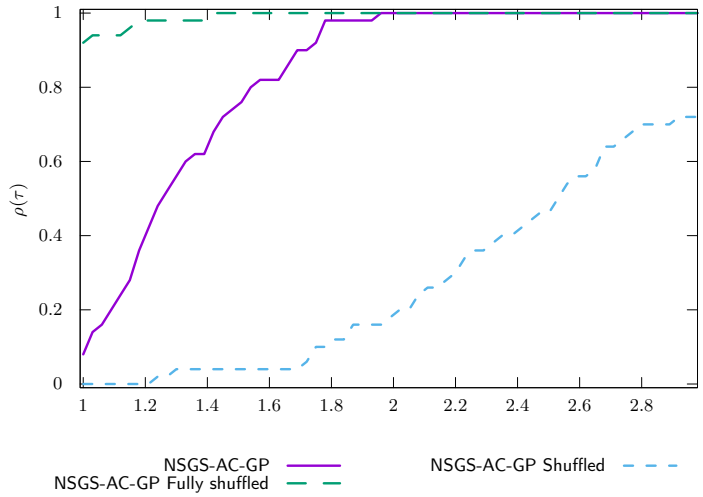


Figure 121: LMGC LowWall FEM time NSGS/Shuffled

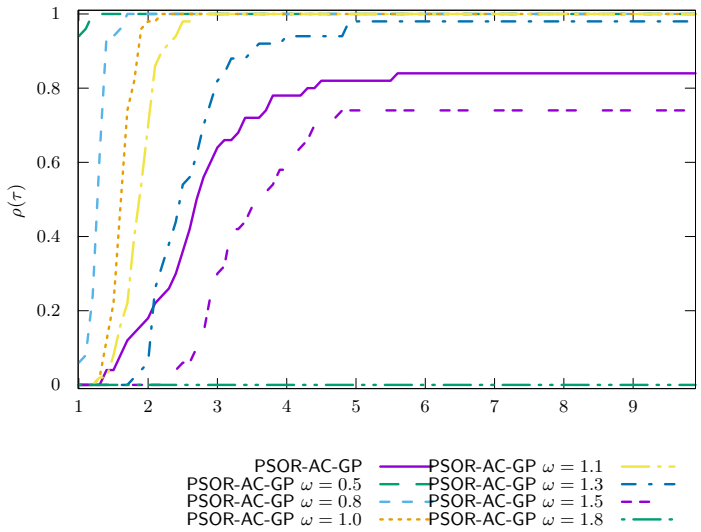


Figure 122: LMGC LowWall FEM time PSOR

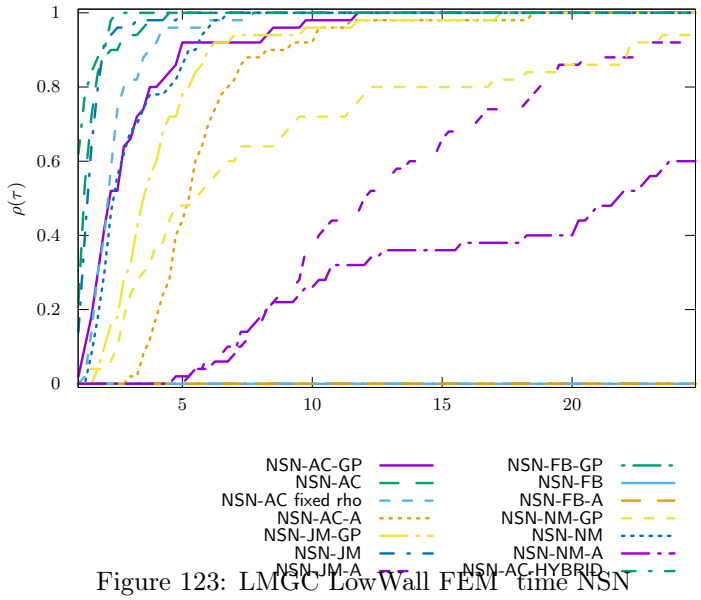


Figure 123: LMGC LowWall FEM time NSN

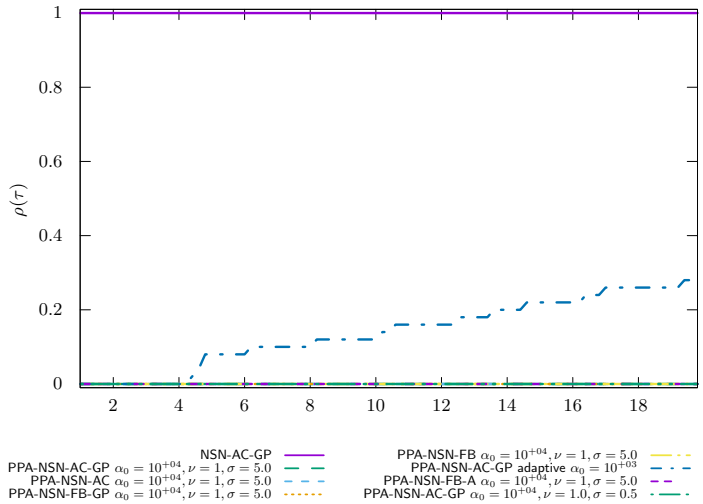


Figure 124: LMGC LowWall FEM time PROX/NSN/InternalSolvers

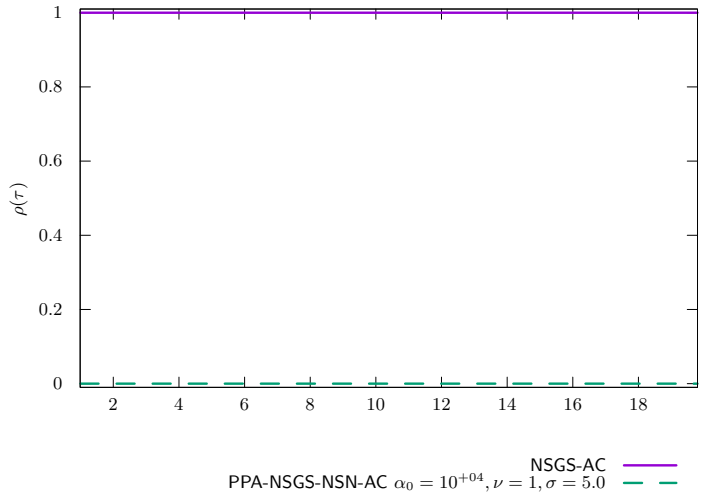


Figure 125: LMGC LowWall FEM time PROX/NSGS/InternalSolvers

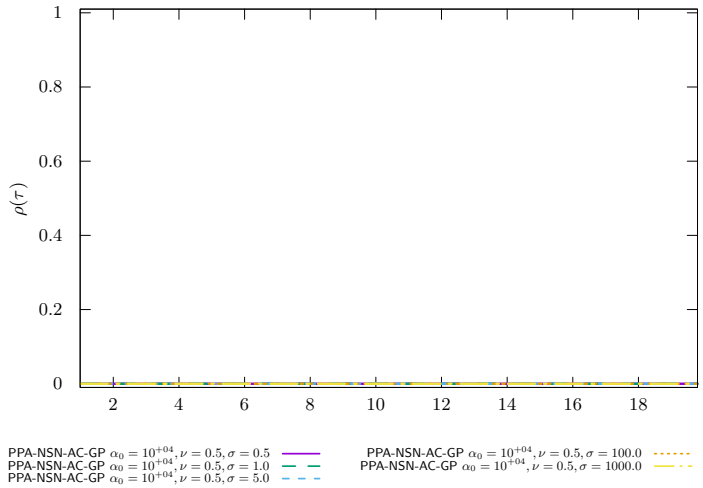


Figure 126: LMGC LowWall FEM time PROX/Parametric studies $\nu = 0.5$

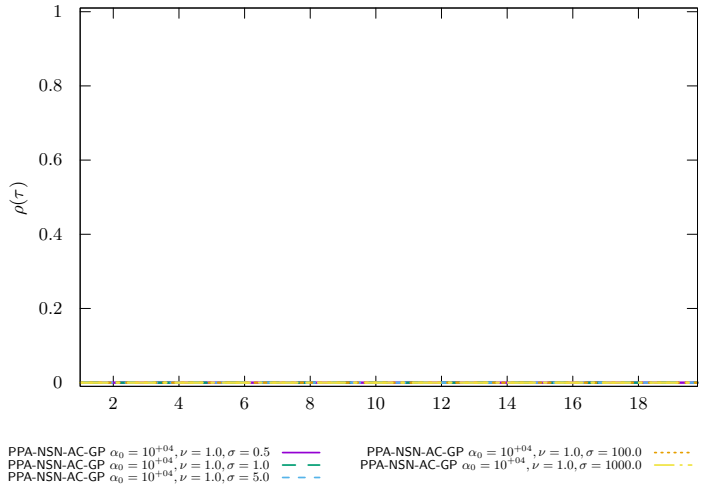


Figure 127: LMGC LowWall FEM time PROX/Parametric studies $\nu = 1.0$

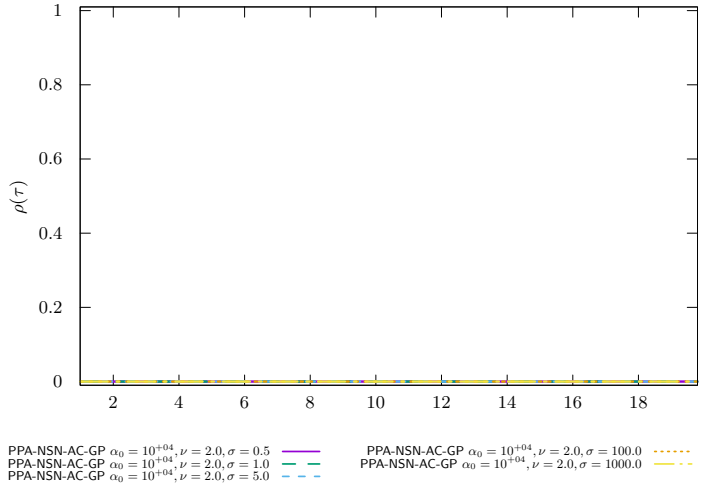


Figure 128: LMGC LowWall FEM time PROX/Parametric studies $\nu = 2.0$

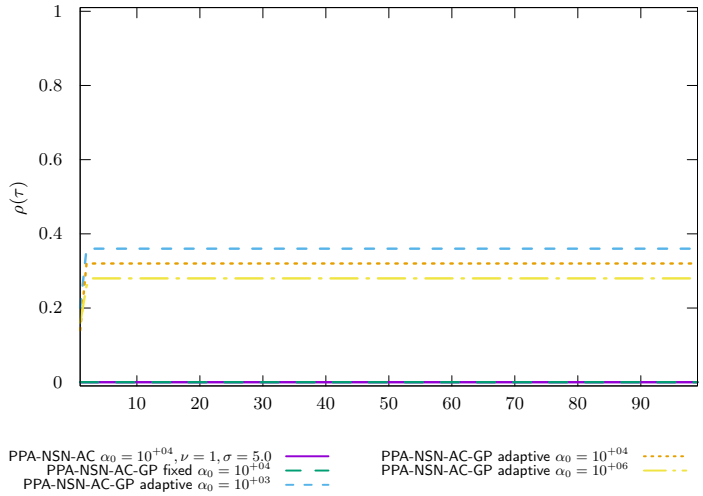


Figure 129: LMGC LowWall FEM time PROX/Regularized problem

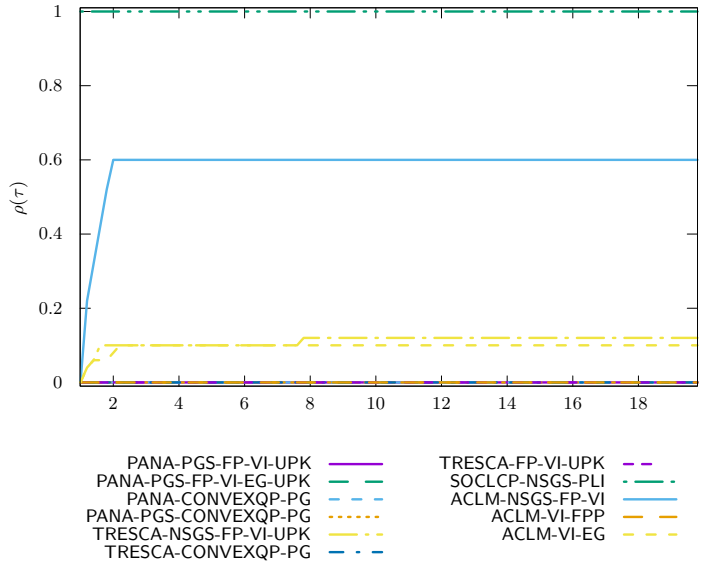


Figure 130: LMGC LowWall FEM time OPTI

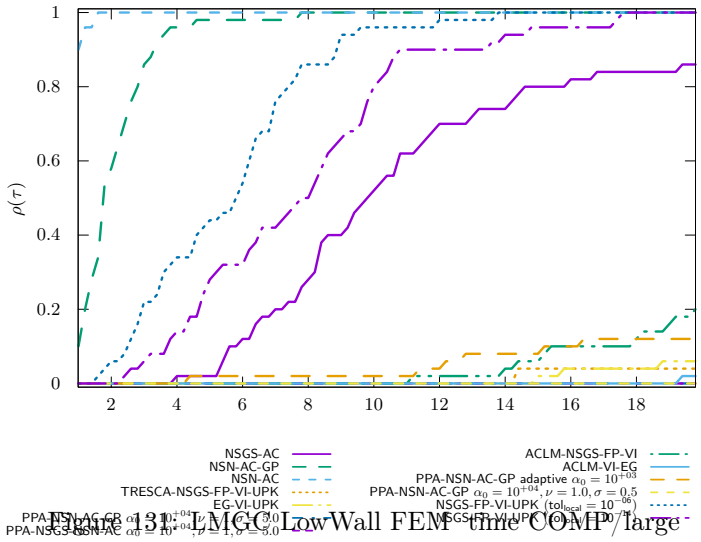


Figure 131: LMGC LowWall FEM time COMP/large

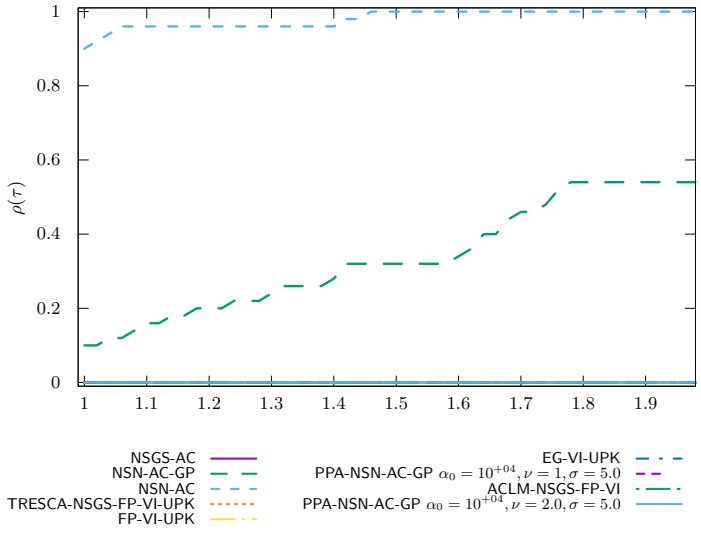


Figure 132: LMGC LowWall FEM time COMP/zoom

LMGC LowWall FEM precision 1.0e-08 timeout 400

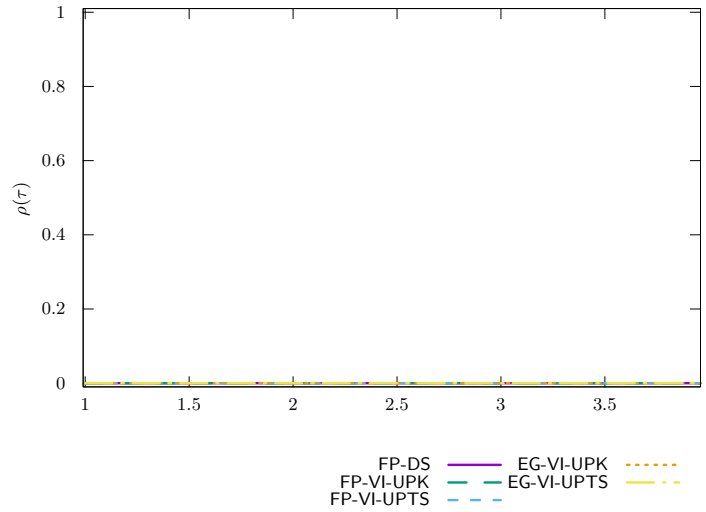


Figure 133: LMGC LowWall FEM time VI/UpdateRule

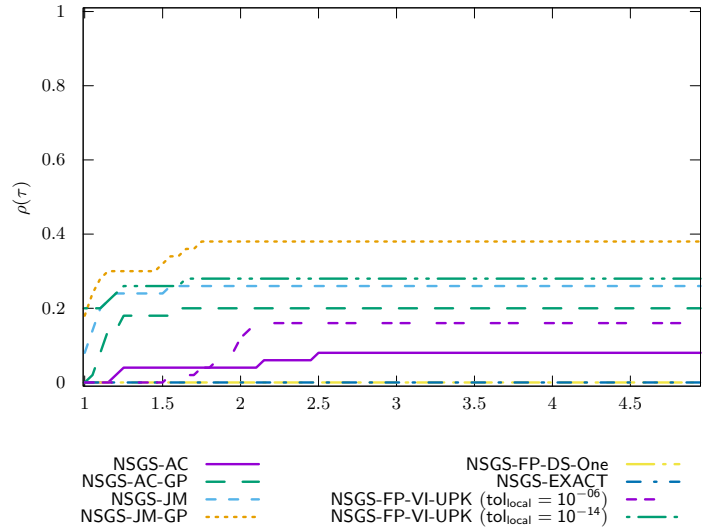


Figure 134: LMGC LowWall FEM time NSGS/LocalSolver

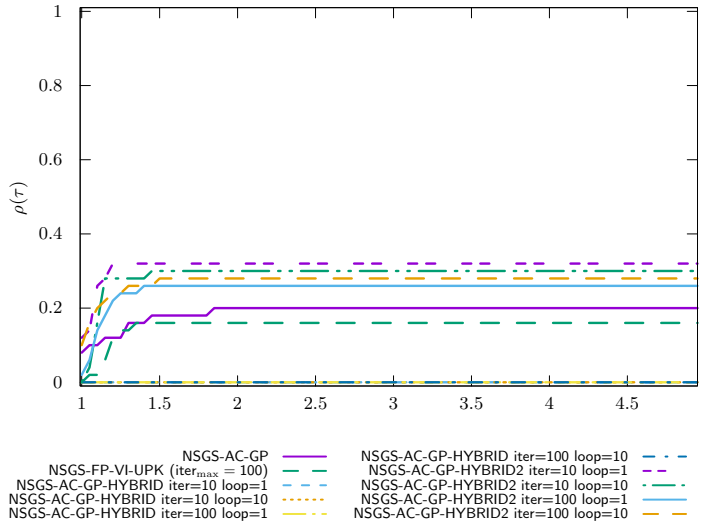


Figure 135: LMGC LowWall FEM time NSGS/LocalSolverHybrid

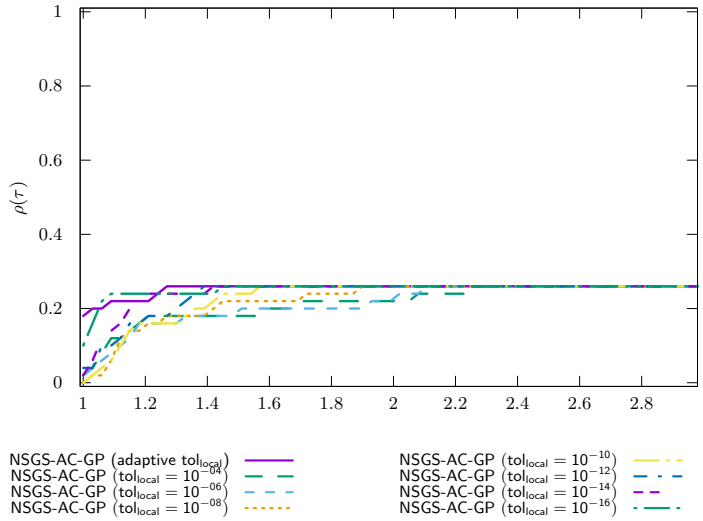


Figure 136: LMGC LowWall FEM time NSGS/LocalTol

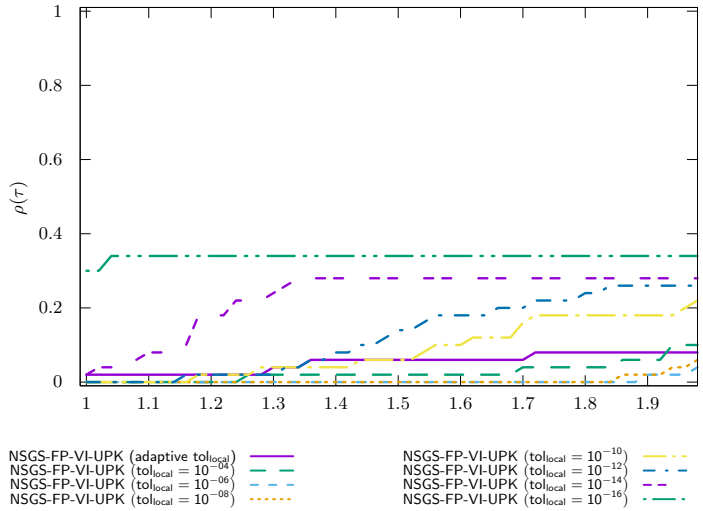


Figure 137: LMGC LowWall FEM time NSGS/LocalTol-VI

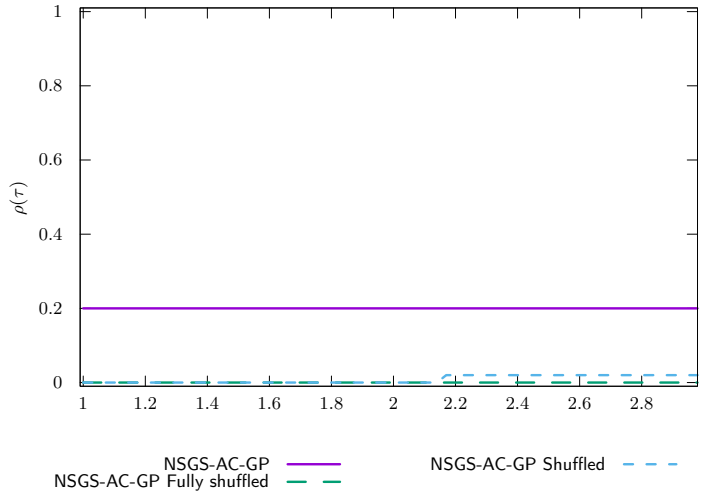


Figure 138: LMGC LowWall FEM time NSGS/Shuffled

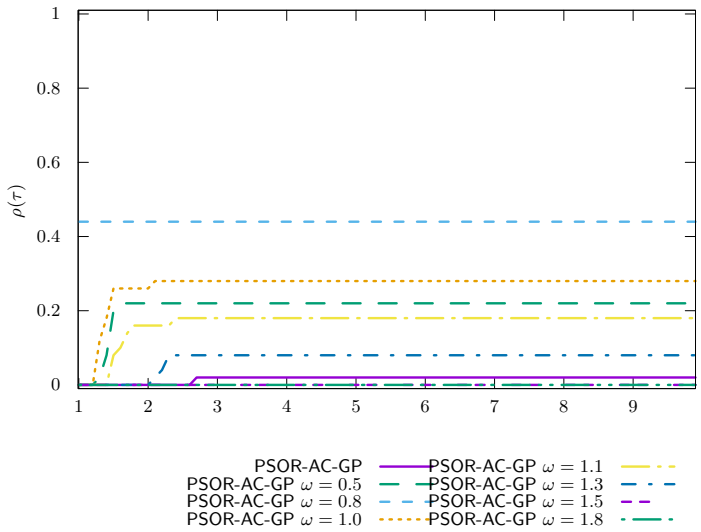


Figure 139: LMGC LowWall FEM time PSOR

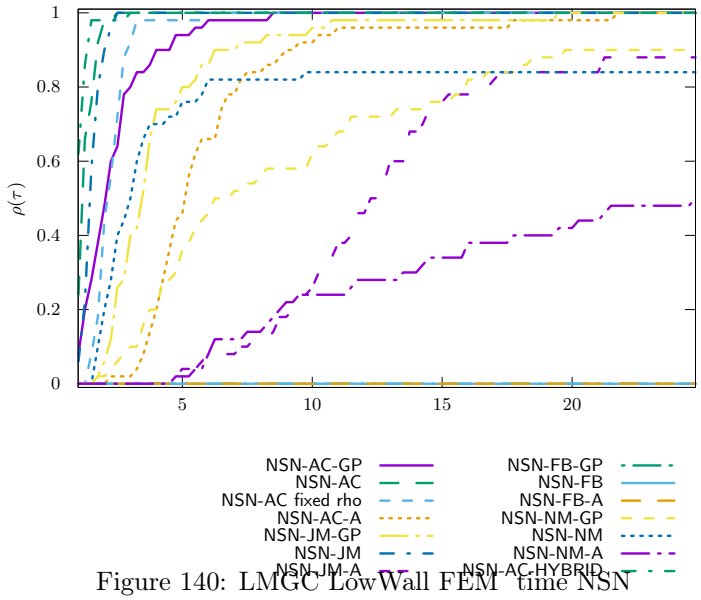


Figure 140: LMGC LowWall FEM time NSN

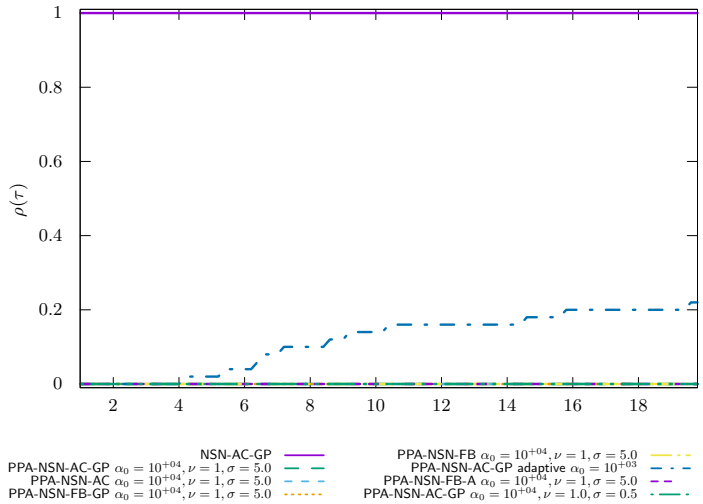


Figure 141: LMGC LowWall FEM time PROX/NSN/InternalSolvers

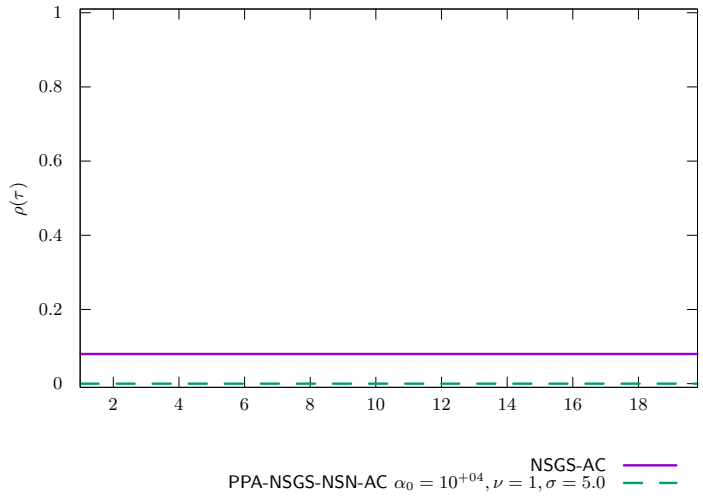


Figure 142: LMGC LowWall FEM time PROX/NSGS/InternalSolvers

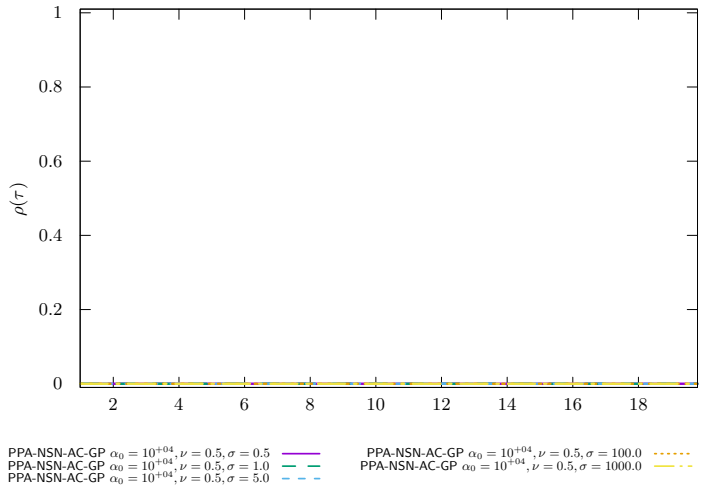


Figure 143: LMGC LowWall FEM time PROX/Parametric studies $\nu = 0.5$

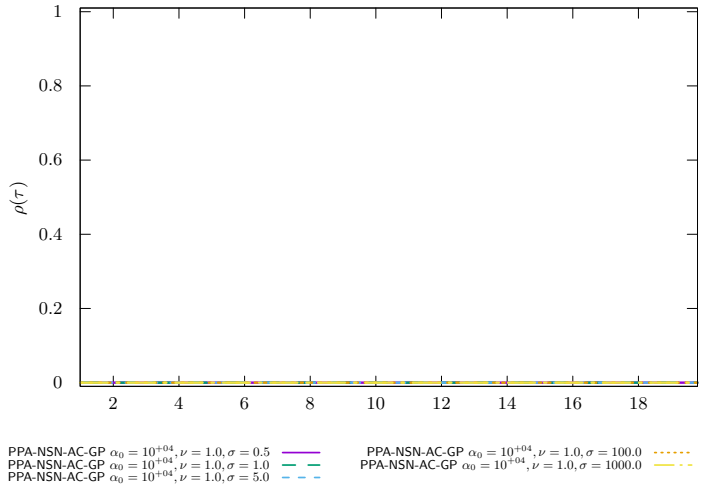


Figure 144: LMGC LowWall FEM time PROX/Parametric studies $\nu = 1.0$

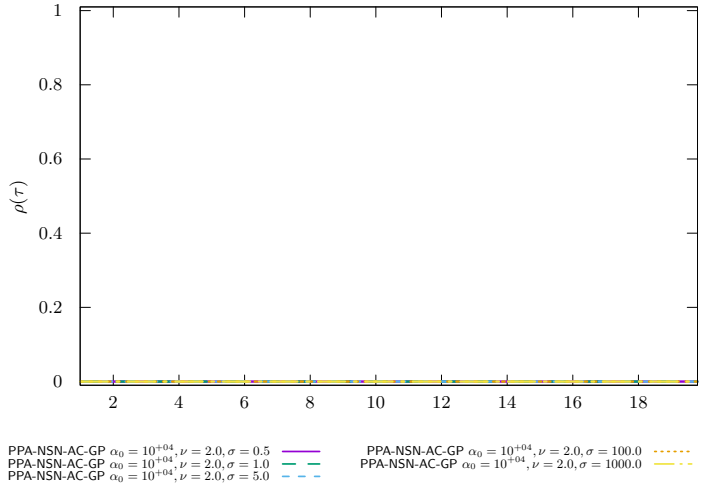


Figure 145: LMGC LowWall FEM time PROX/Parametric studies $\nu = 2.0$

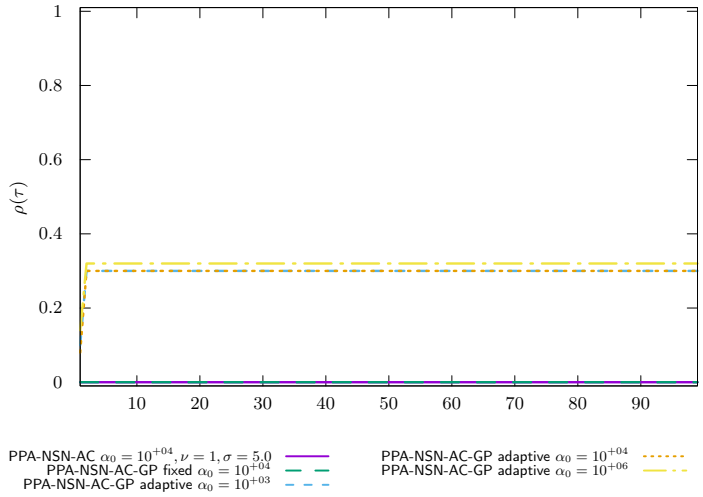


Figure 146: LMGC LowWall FEM time PROX/Regularized problem

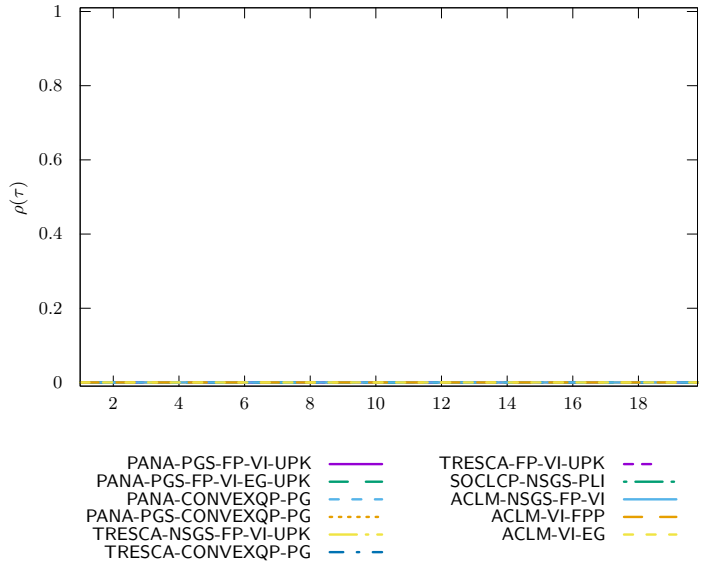


Figure 147: LMGC LowWall FEM time OPTI

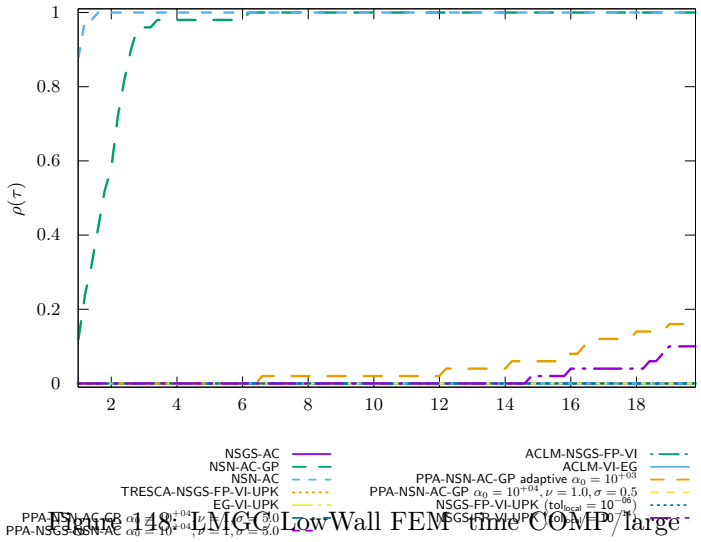


Figure 148: LMGC LowWall FEM time COMP/large

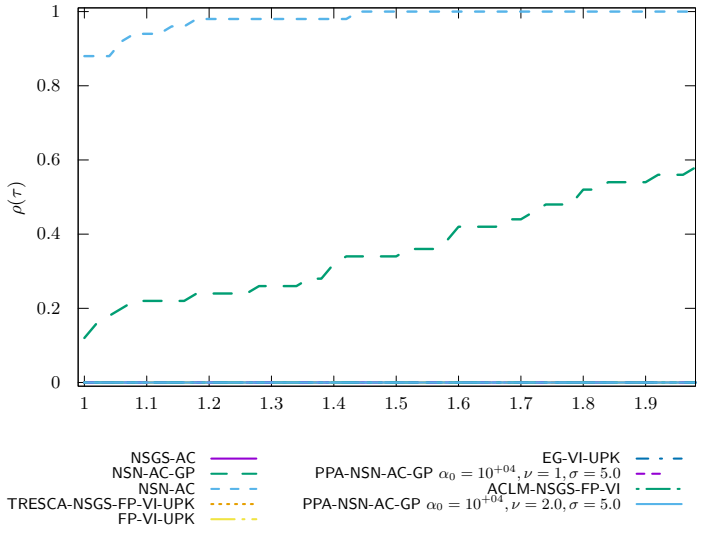


Figure 149: LMGC LowWall FEM time COMP/zoom

LMGC Cubes H8 precision 1.0e-04 timeout 100

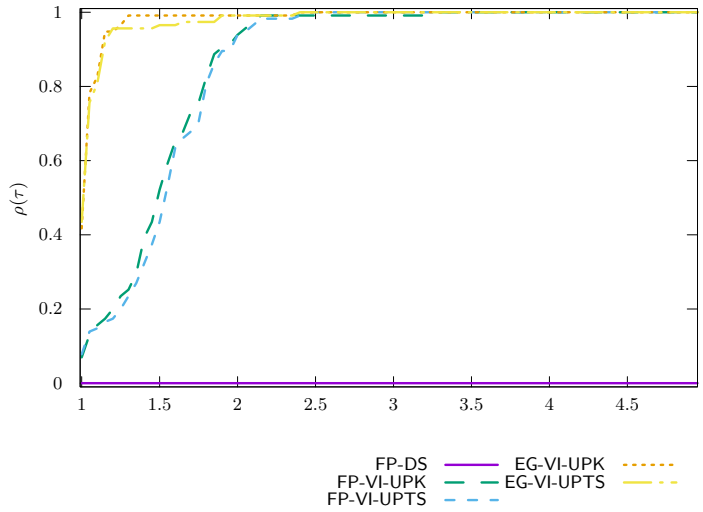


Figure 150: LMGC Cubes H8 time VI/UpdateRule

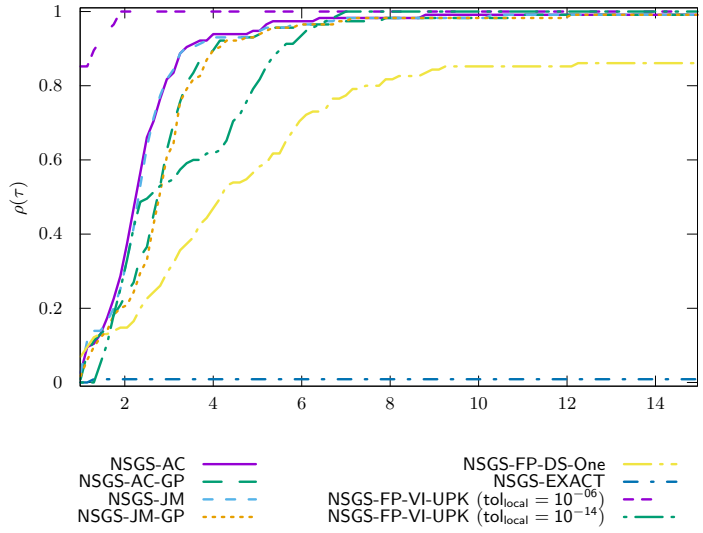


Figure 151: LMGC Cubes H8 time NSGS/LocalSolver

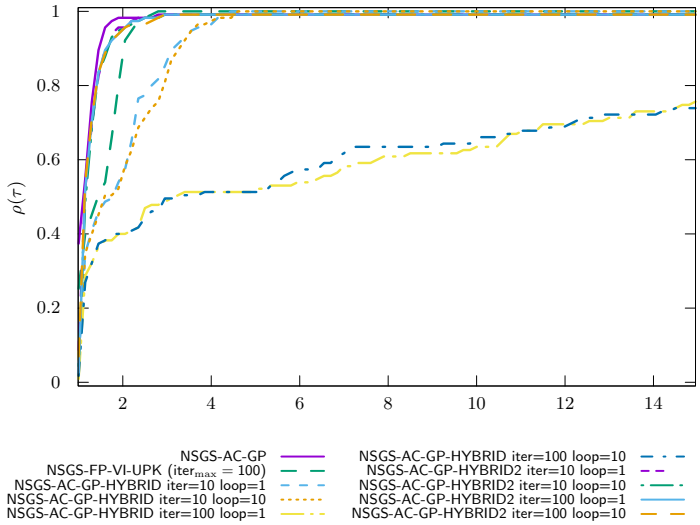


Figure 152: LMGC Cubes H8 time NSGS/LocalSolverHybrid

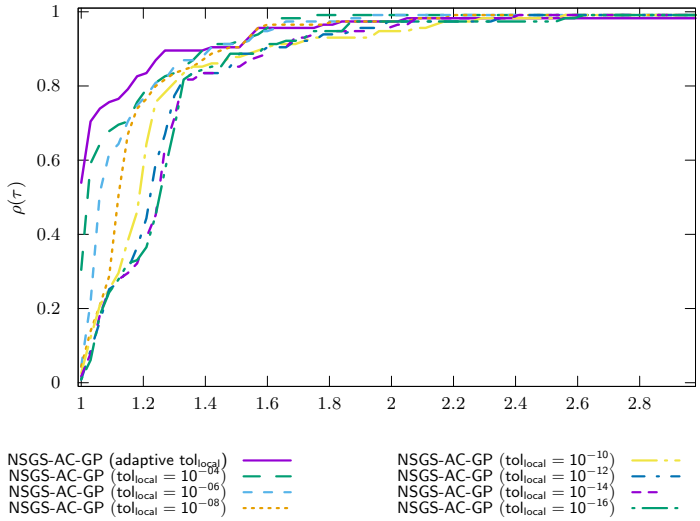


Figure 153: LMGC Cubes H8 time NSGS/LocalTol

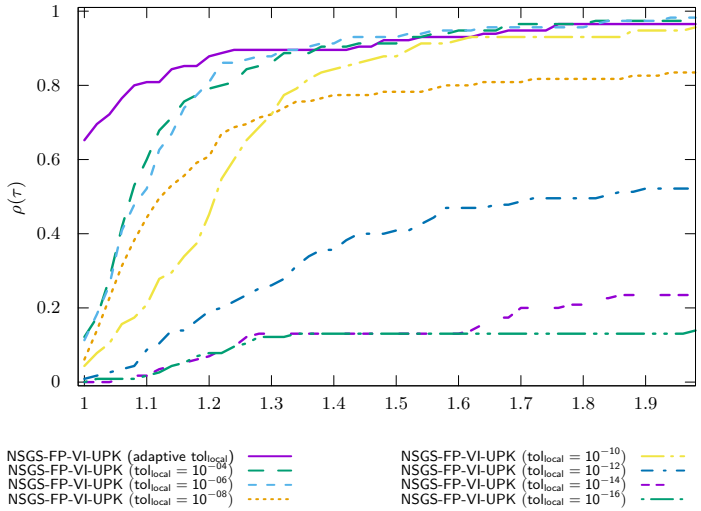


Figure 154: LMGC Cubes H8 time NSGS/LocalTol-VI

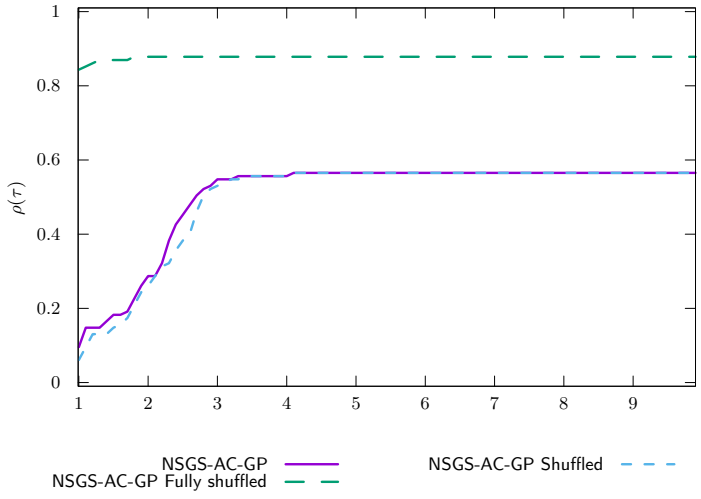


Figure 155: LMGC Cubes H8 time NSGS/Shuffled

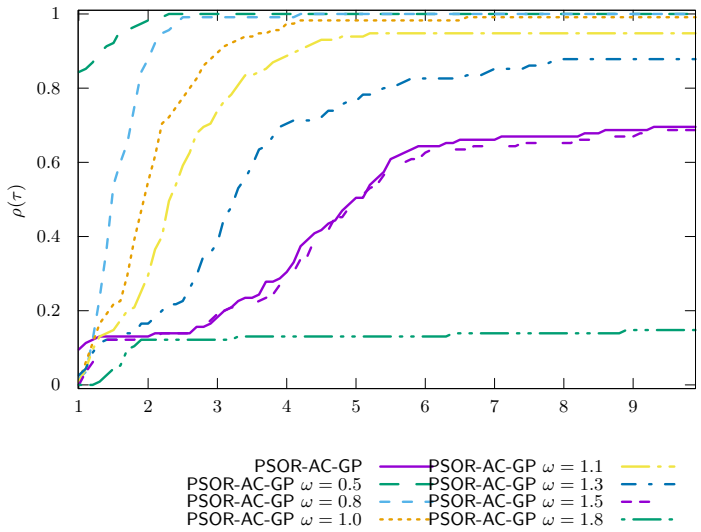


Figure 156: LMGC Cubes H8 time PSOR

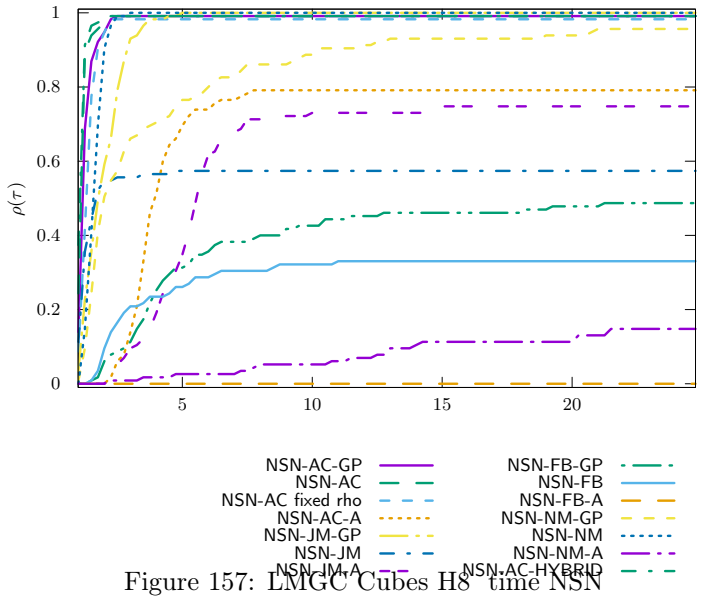


Figure 157: LMGC Cubes H8 time NSN

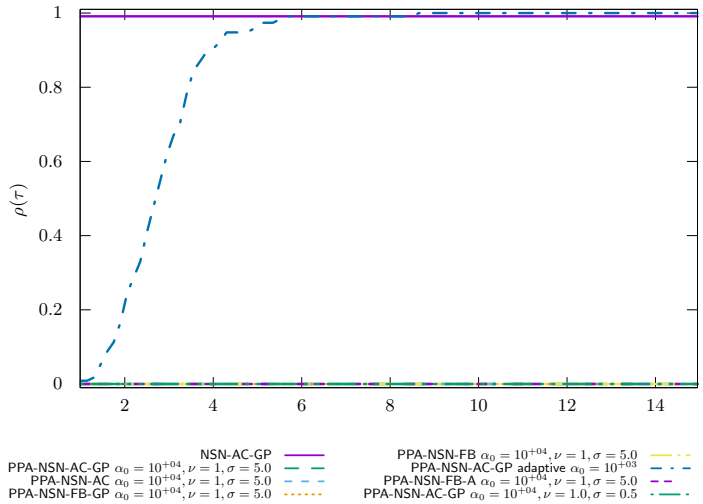


Figure 158: LMGC Cubes H8 time PROX/NSN/InternalSolvers

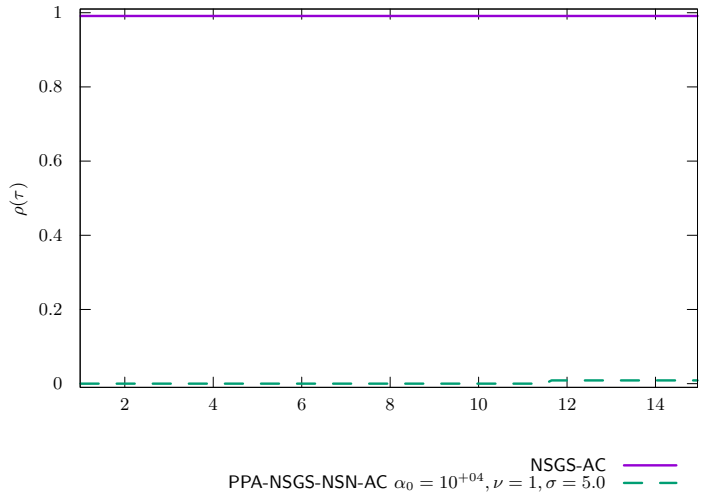


Figure 159: LMGC Cubes H8 time PROX/NSGS/InternalSolvers

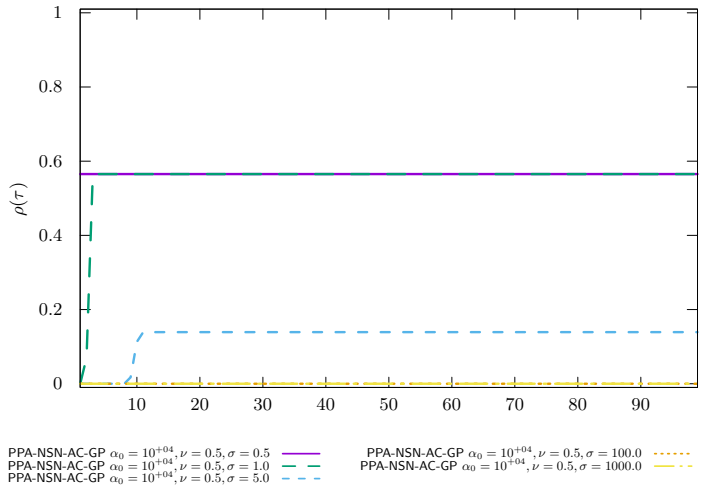


Figure 160: LMGC Cubes H8 time PROX/Parametric studies $\nu = 0.5$

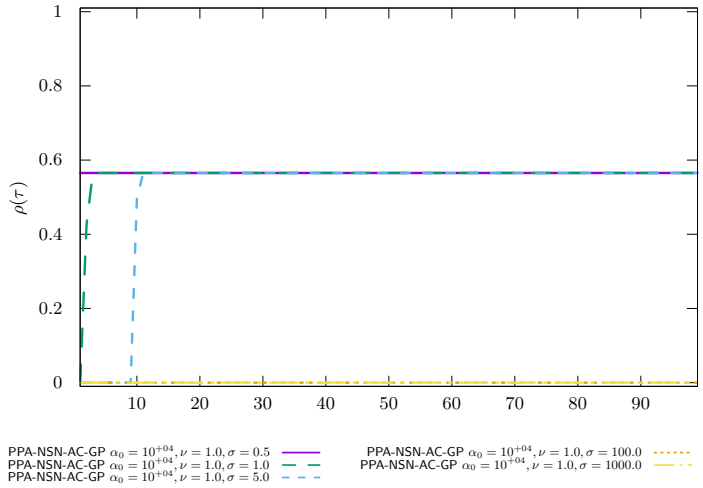


Figure 161: LMGC Cubes H8 time PROX/Parametric studies $\nu = 1.0$

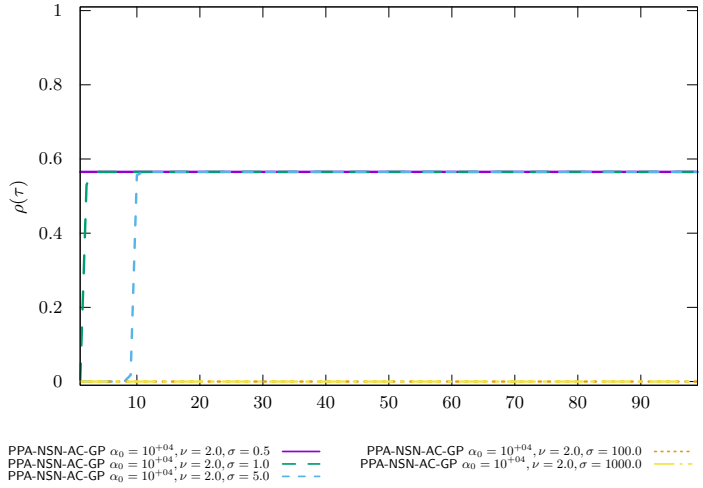


Figure 162: LMGC Cubes H8 time PROX/Parametric studies $\nu = 2.0$

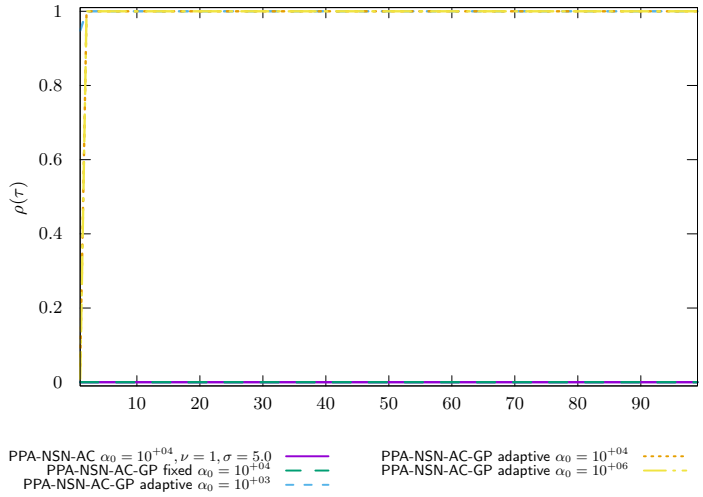


Figure 163: LMGC Cubes H8 time PROX/Regularized problem

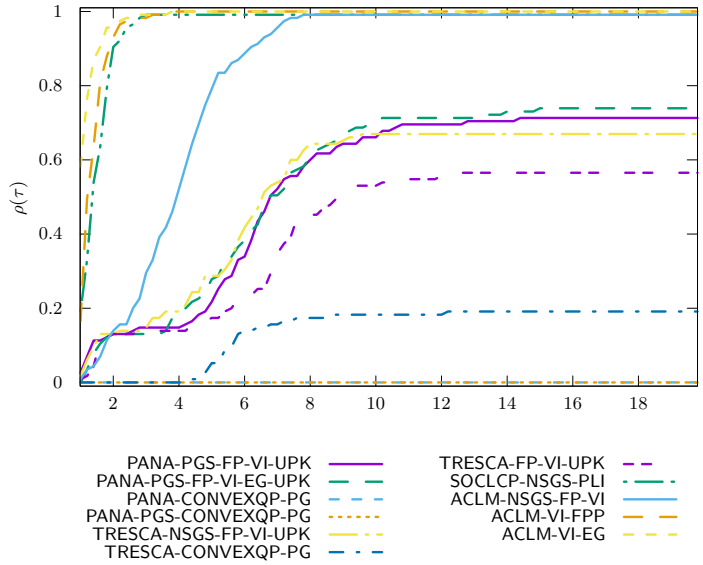


Figure 164: LMGC Cubes H8 time OPTI

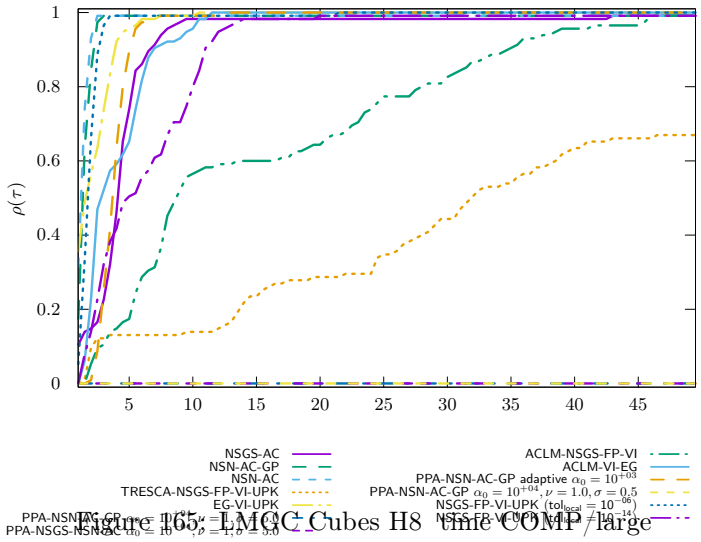


Figure 165: LMGC Cubes H8 time COMP/large

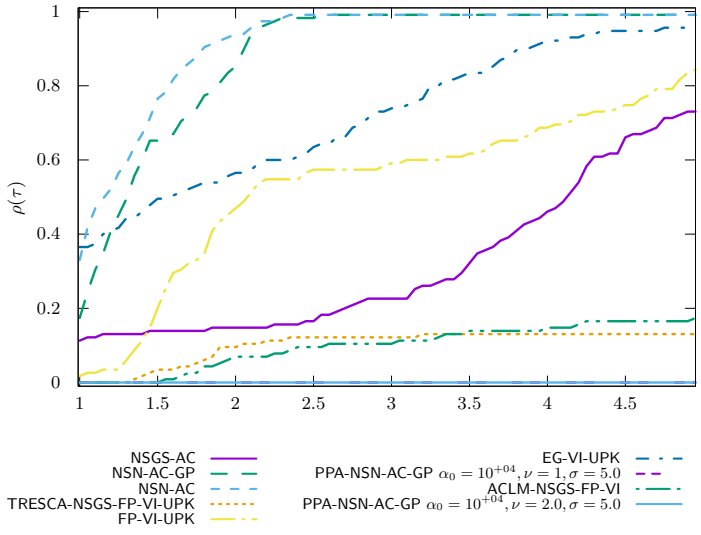


Figure 166: LMGC Cubes H8 time COMP/zoom

Capsules precision 1.0e-08 timeout 50

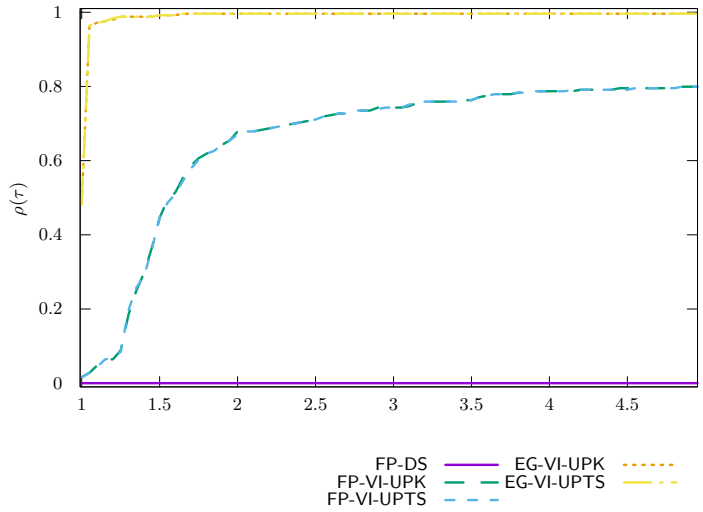


Figure 167: Capsules time VI/UpdateRule

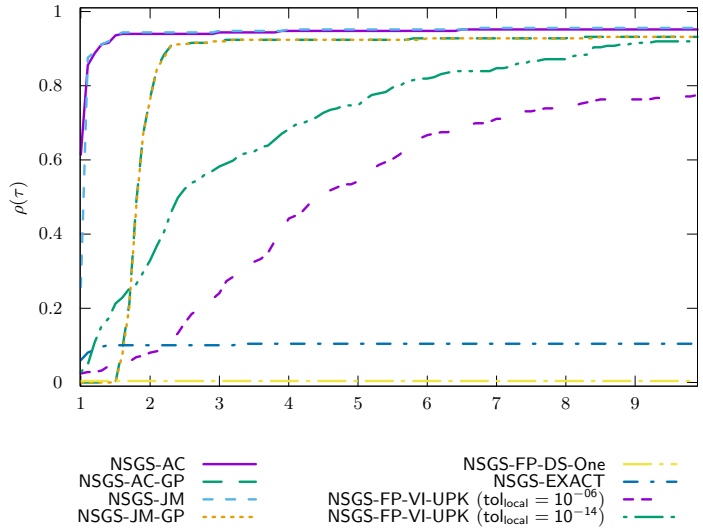


Figure 168: Capsules time NSGS/LocalSolver

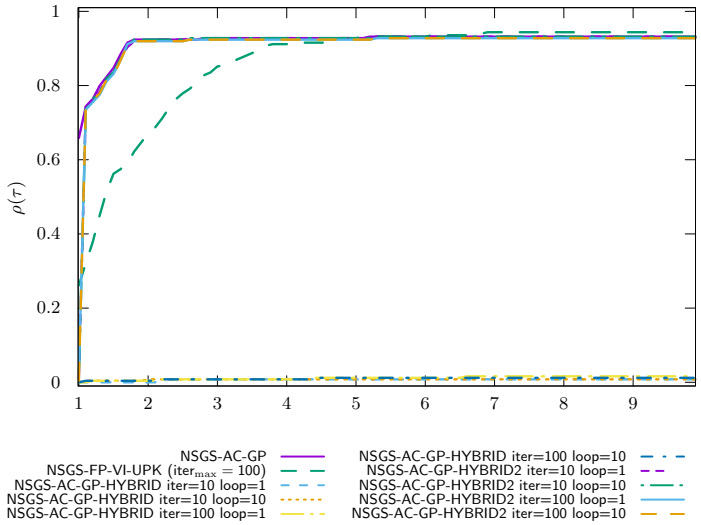


Figure 169: Capsules time NSGS/LocalSolverHybrid

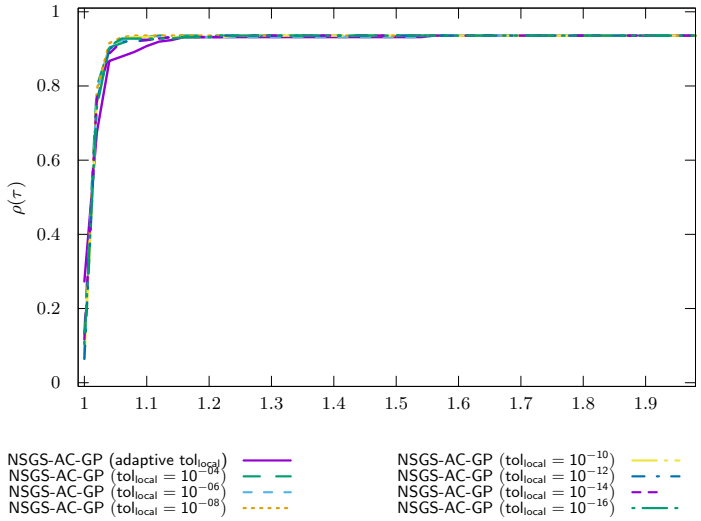


Figure 170: Capsules time NSGS/LocalTol

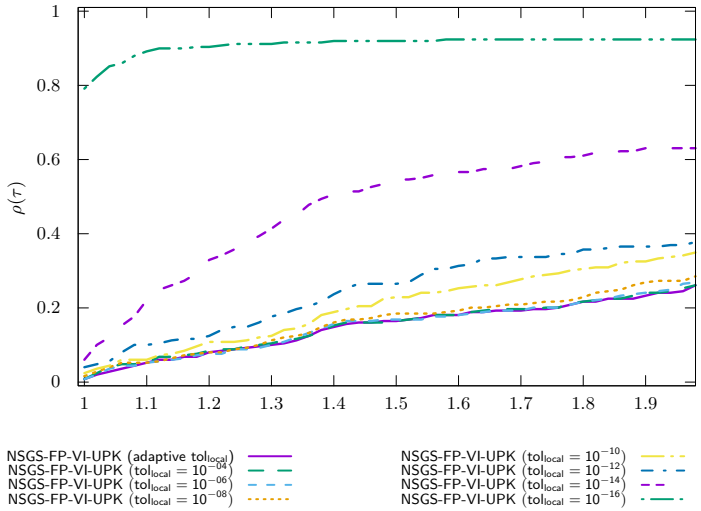


Figure 171: Capsules time NSGS/LocalTol-VI

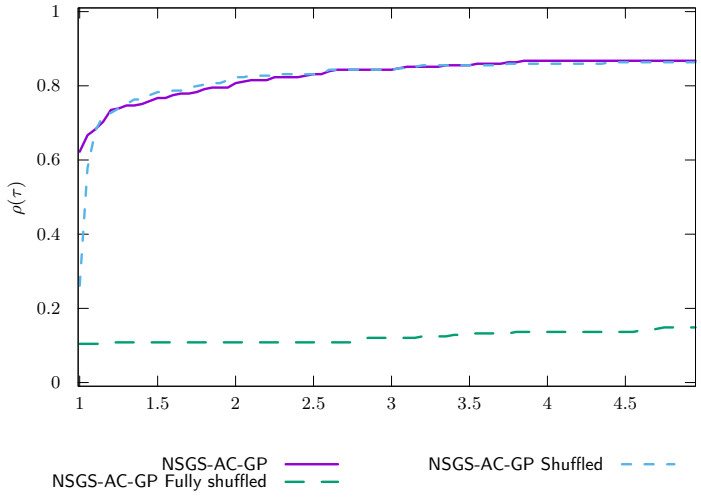


Figure 172: Capsules time NSGS/Shuffled

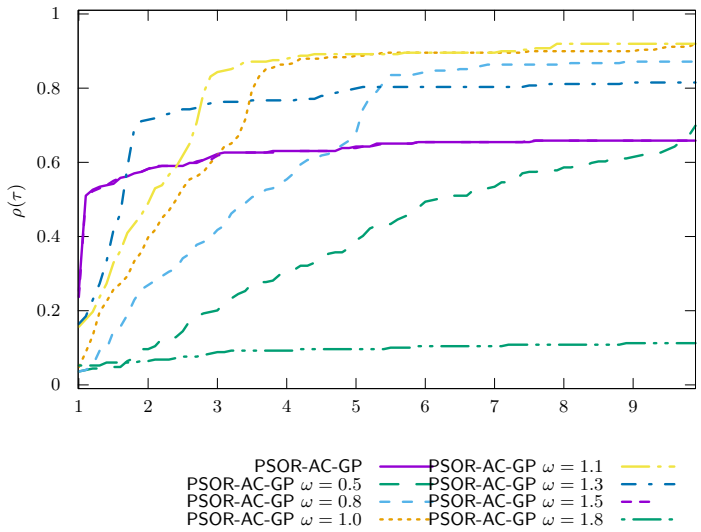


Figure 173: Capsules time PSOR

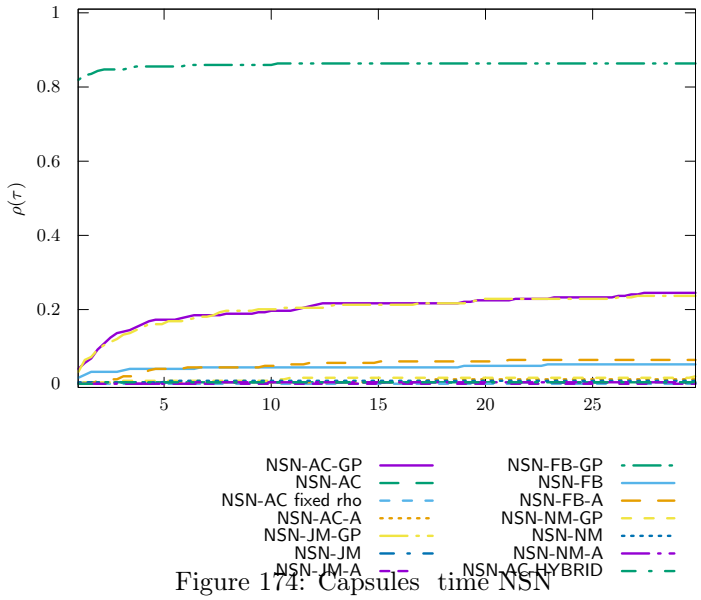


Figure 174: Capsules time NSN

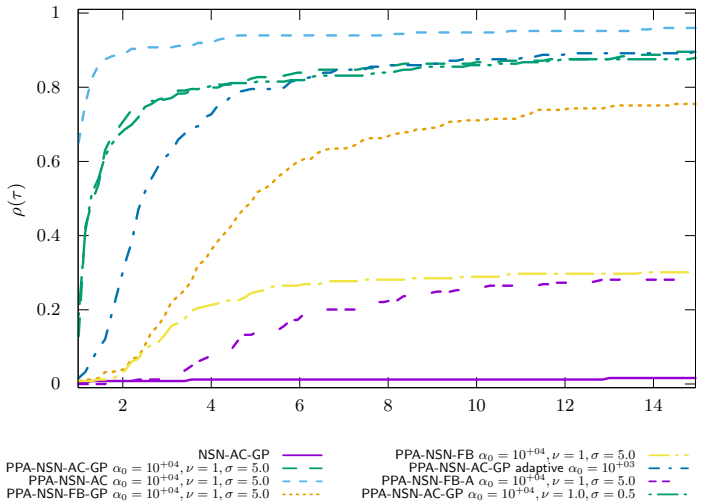


Figure 175: Capsules time PROX/NSN/InternalSolvers

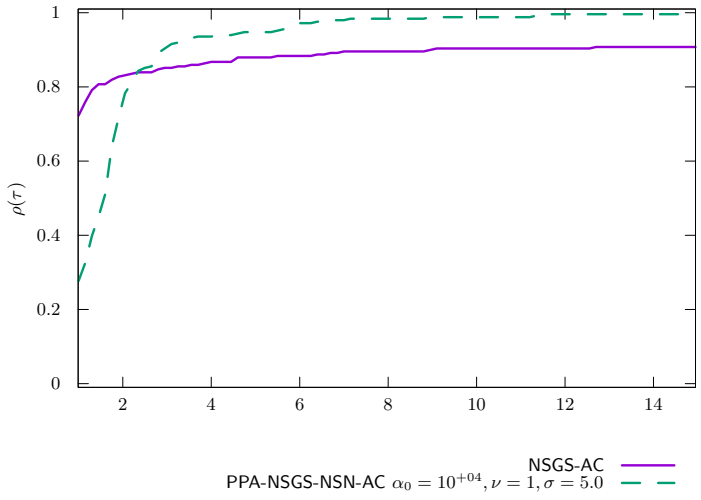


Figure 176: Capsules time PROX/NSGS/InternalSolvers

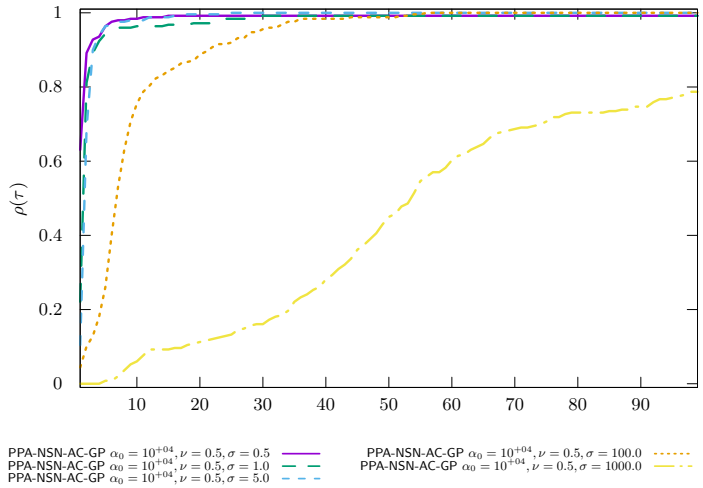


Figure 177: Capsules time PROX/Parametric studies $\nu = 0.5$

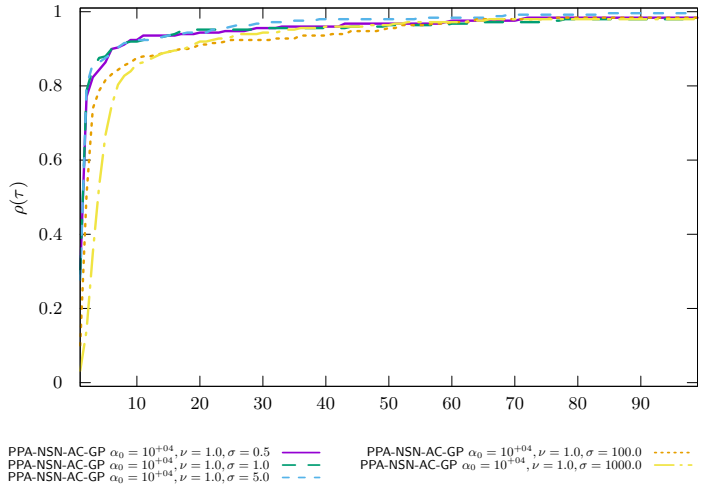


Figure 178: Capsules time PROX/Parametric studies $\nu = 1.0$

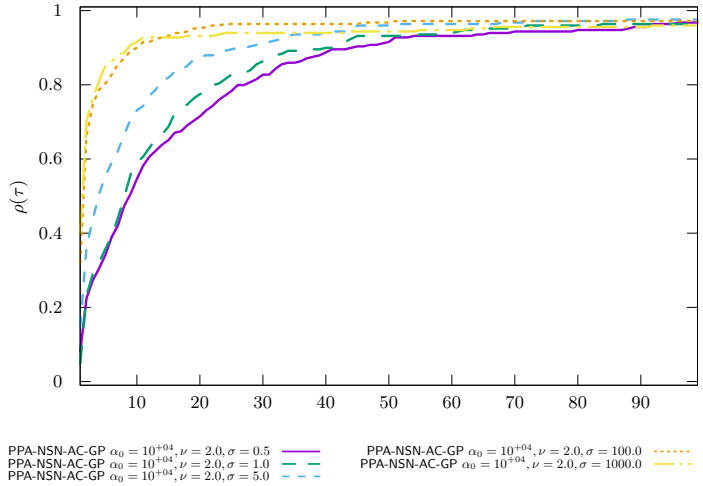


Figure 179: Capsules time PROX/Parametric studies $\nu = 2.0$

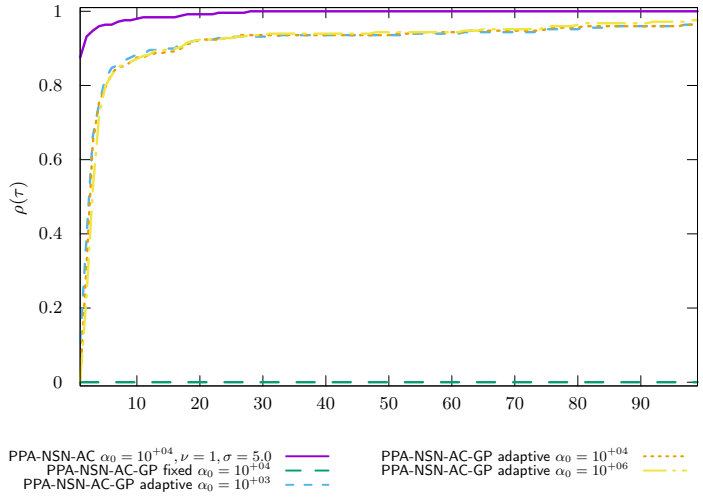


Figure 180: Capsules time PROX/Regularized problem

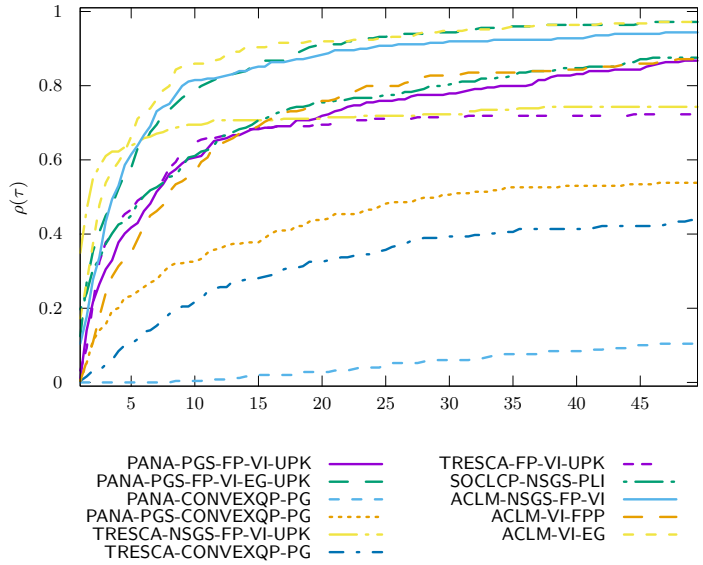


Figure 181: Capsules time OPTI

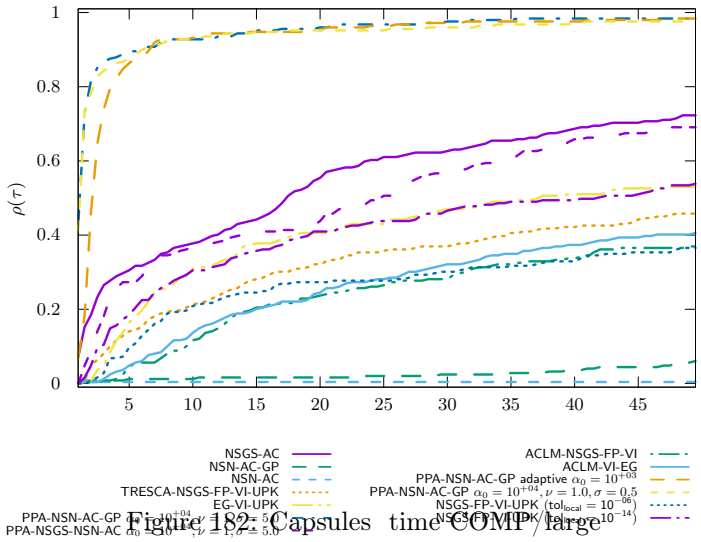


Figure 182: Capsules time COMP / large

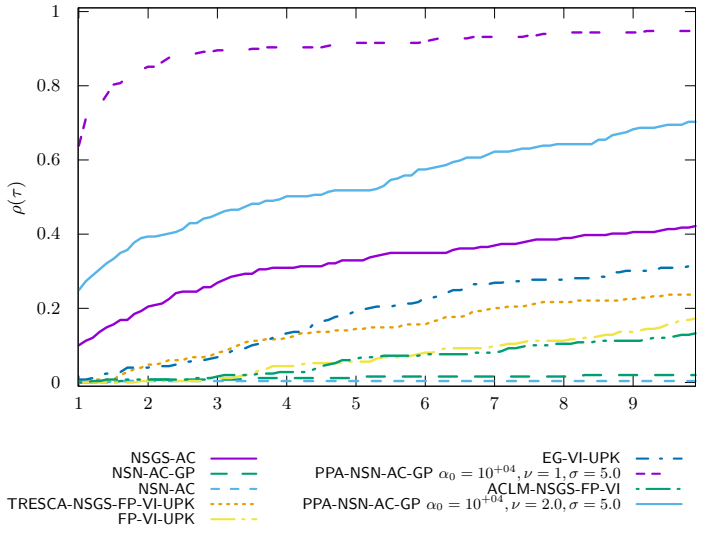


Figure 183: Capsules time COMP/zoom

Chain precision 1.0e-08 timeout 50

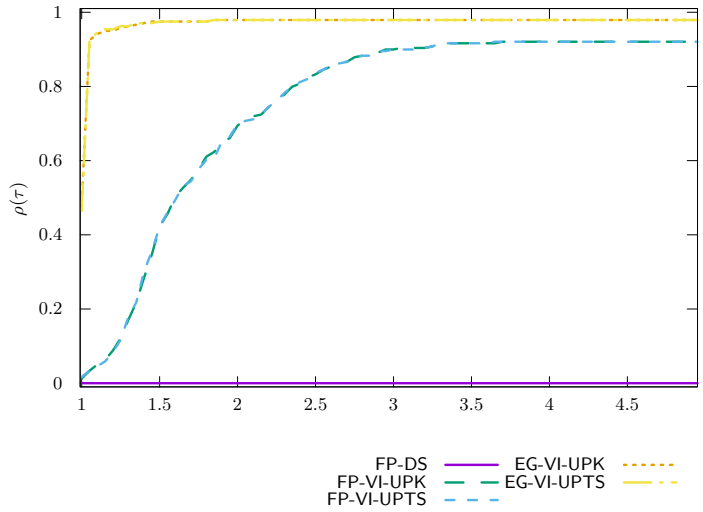


Figure 184: Chain time VI/UpdateRule

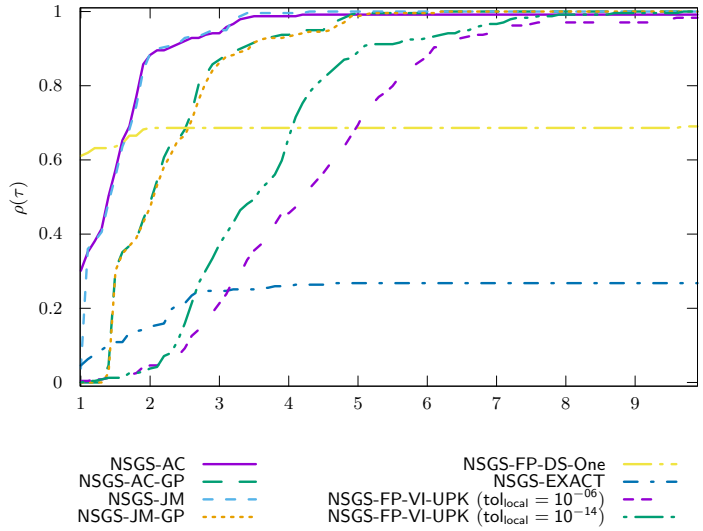


Figure 185: Chain time NSGS/LocalSolver

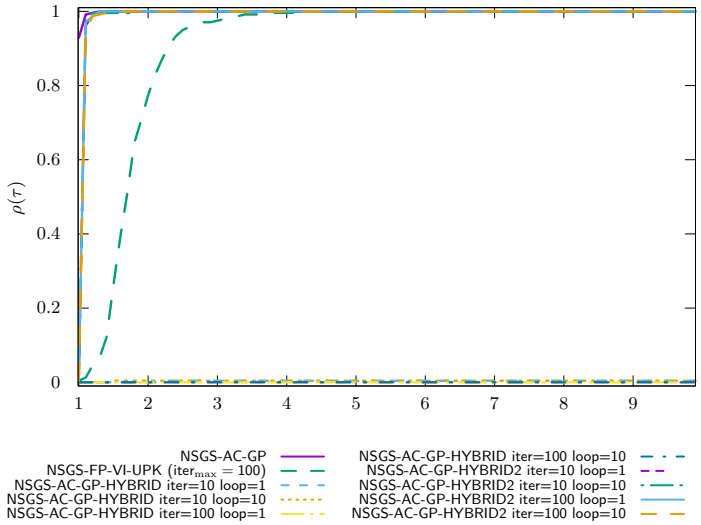


Figure 186: Chain time NSGS/LocalSolverHybrid

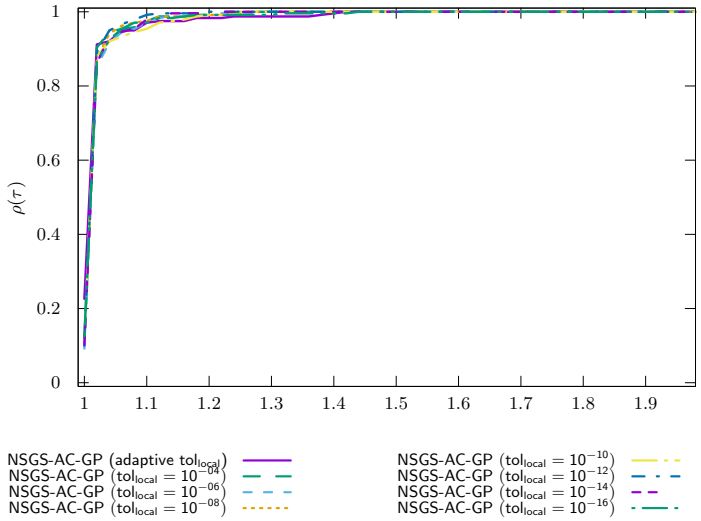


Figure 187: Chain time NSGS/LocalTol

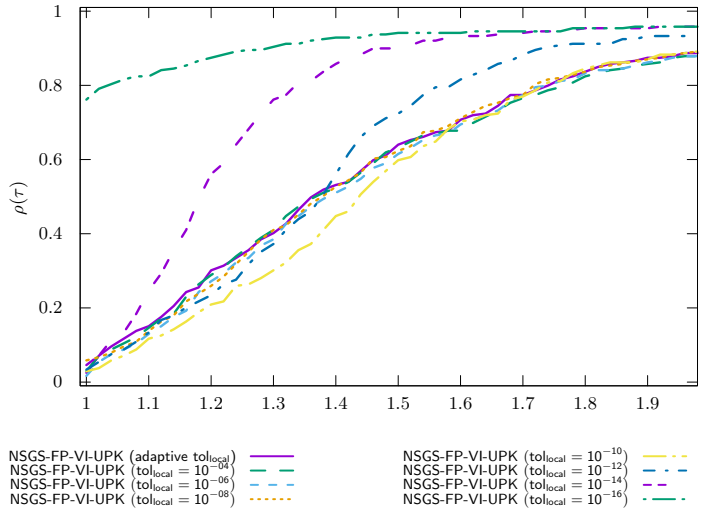


Figure 188: Chain time NSGS/LocalTol-VI

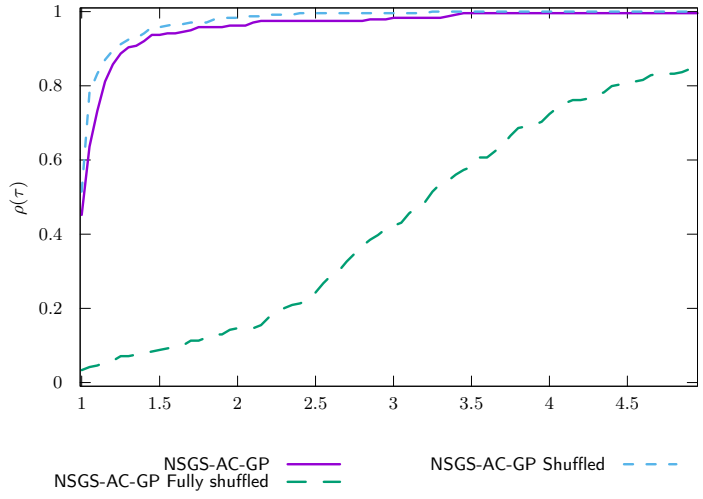


Figure 189: Chain time NSGS/Shuffled

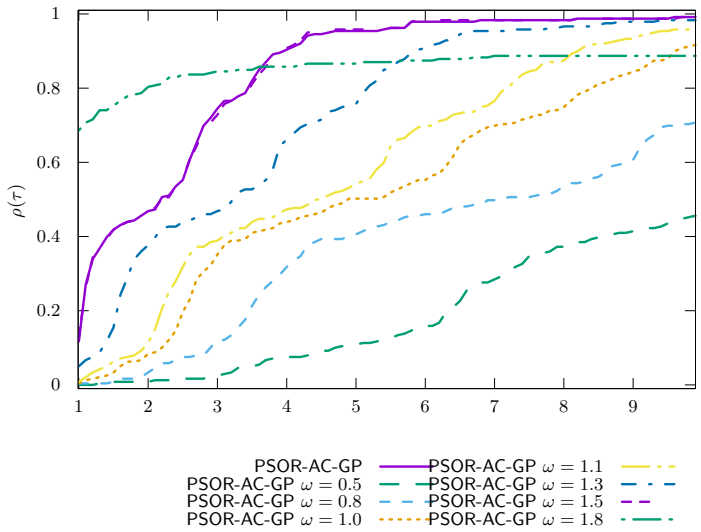


Figure 190: Chain time PSOR

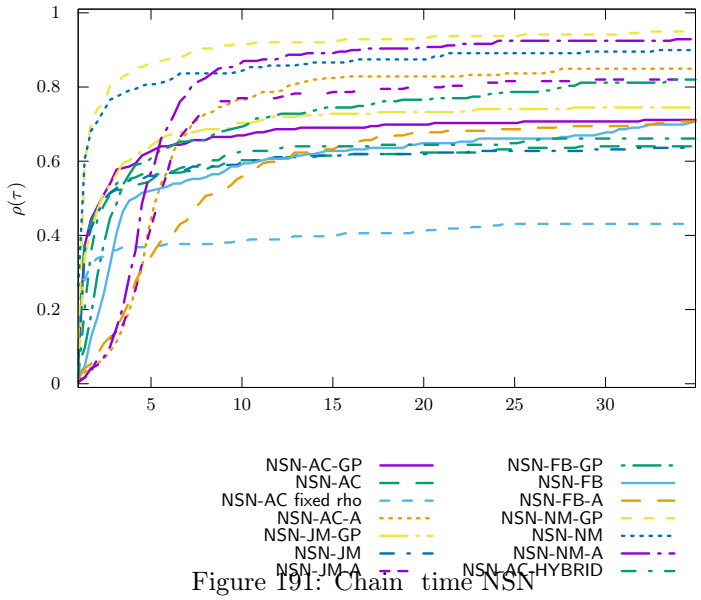


Figure 191: Chain time NSN

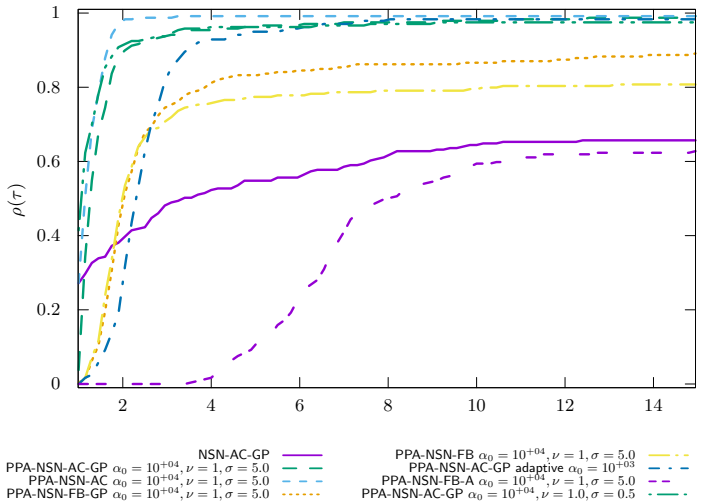


Figure 192: Chain time PROX/NSN/InternalSolvers

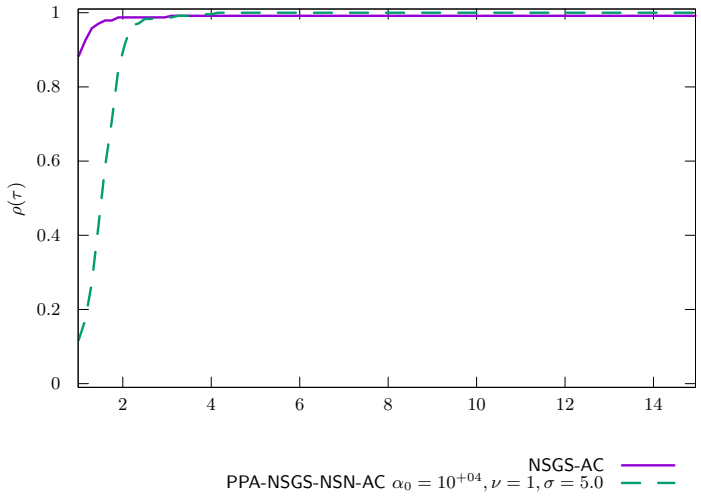


Figure 193: Chain time PROX/NSGS/InternalSolvers

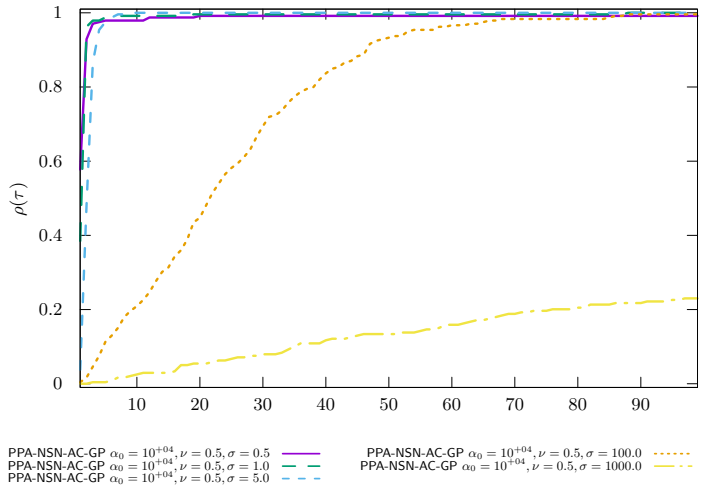


Figure 194: Chain time PROX/Parametric studies $\nu = 0.5$

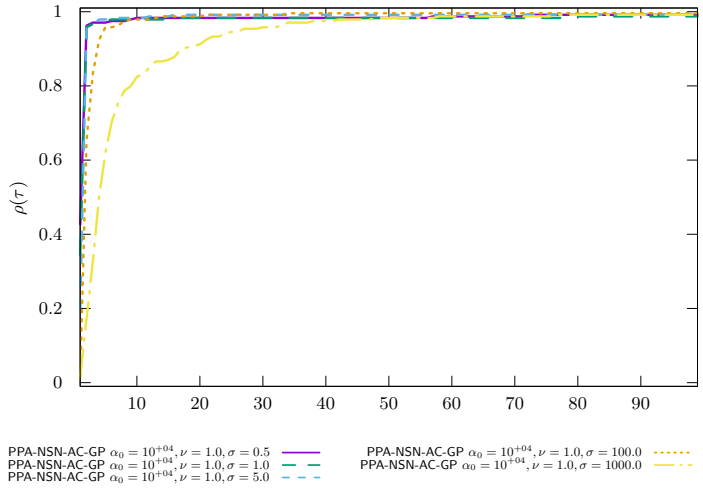


Figure 195: Chain time PROX/Parametric studies $\nu = 1.0$

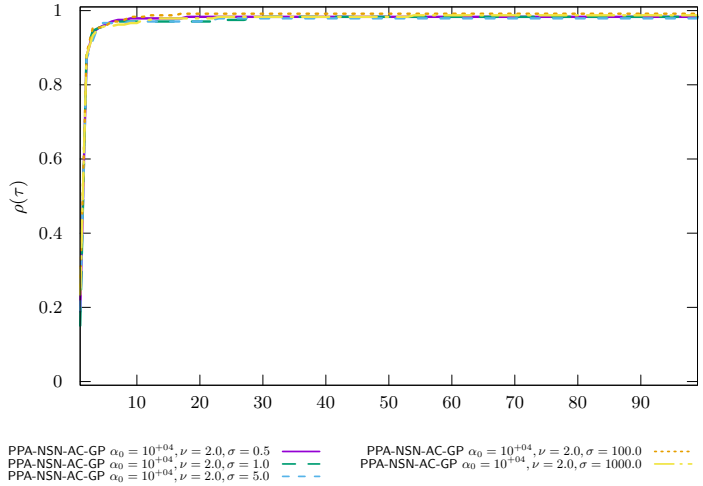


Figure 196: Chain time PROX/Parametric studies $\nu = 2.0$

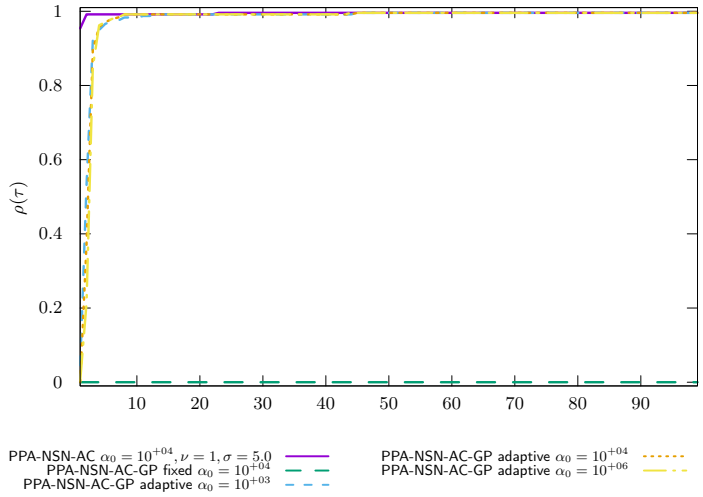


Figure 197: Chain time PROX/Regularized problem

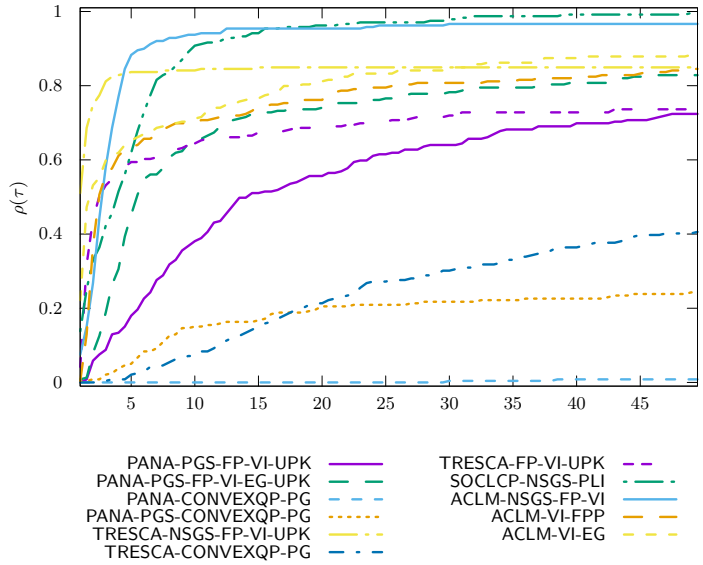


Figure 198: Chain time OPTI

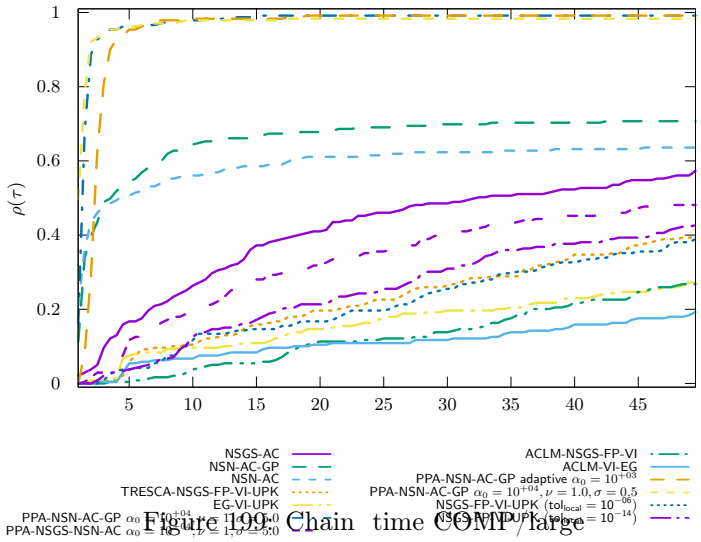


Figure 199: Chain time COMP/large

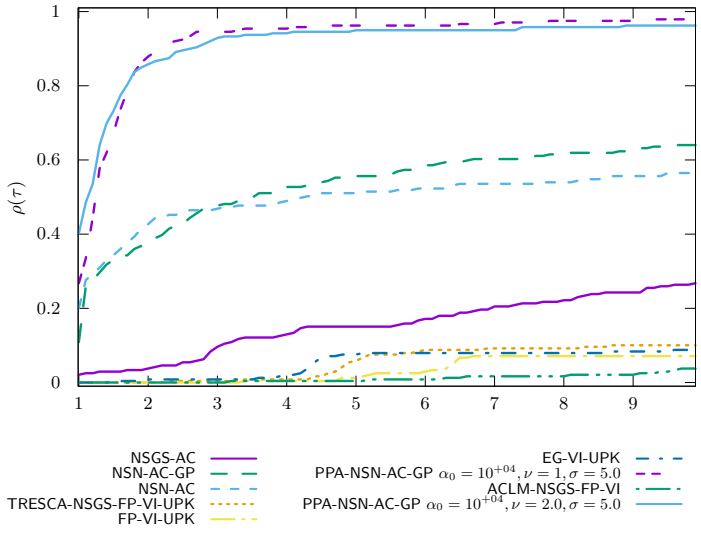


Figure 200: Chain time COMP/zoom

BoxesStack1 precision 1.0e-08 timeout 100

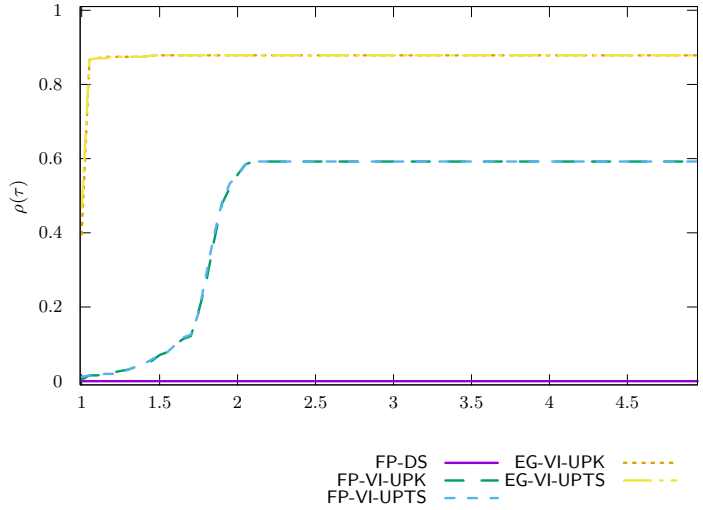


Figure 201: BoxesStack1 time VI/UpdateRule

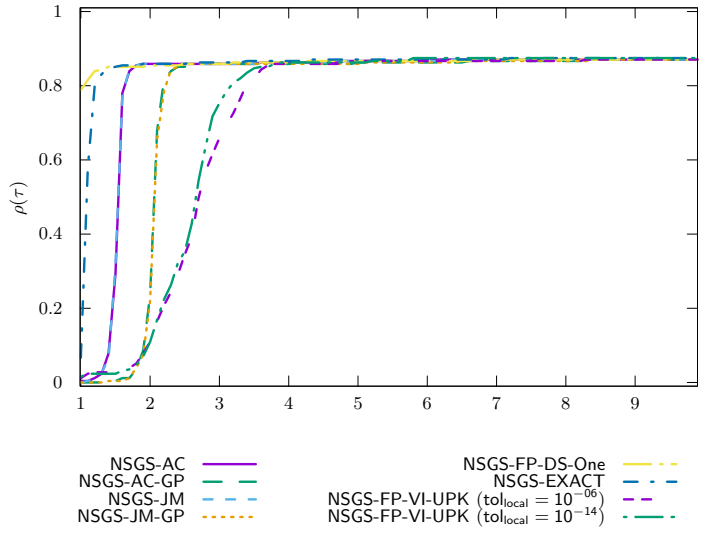


Figure 202: BoxesStack1 time NSGS/LocalSolver

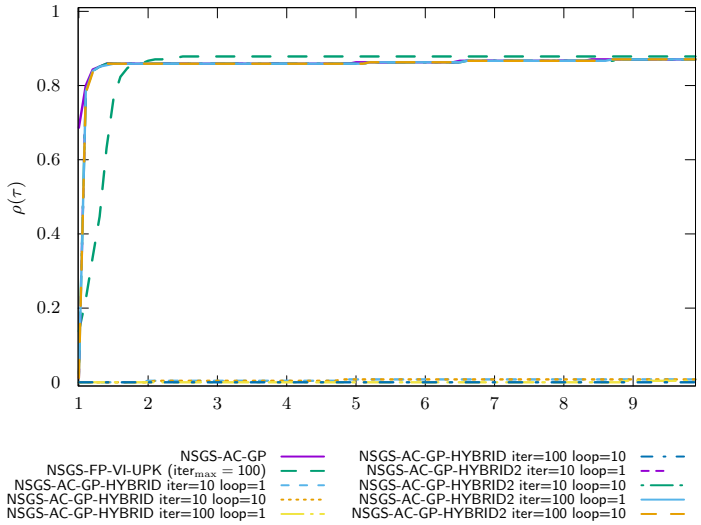


Figure 203: BoxesStack1 time NSGS/LocalSolverHybrid

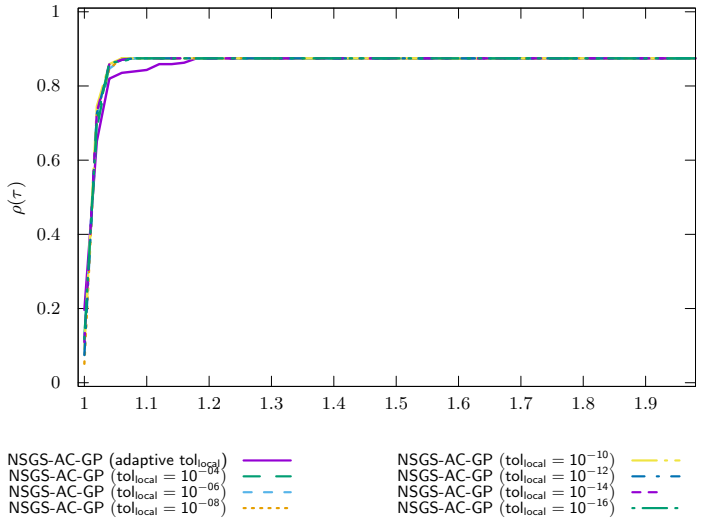


Figure 204: BoxesStack1 time NSGS/LocalTol

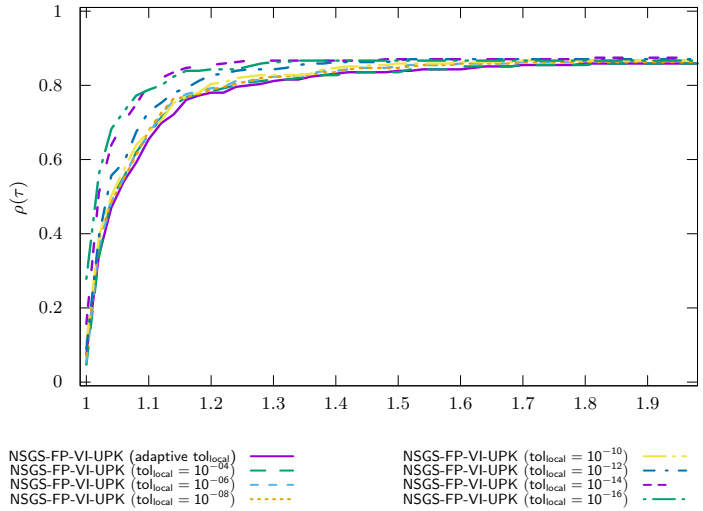


Figure 205: BoxesStack1 time NSGS/LocalTol-VI

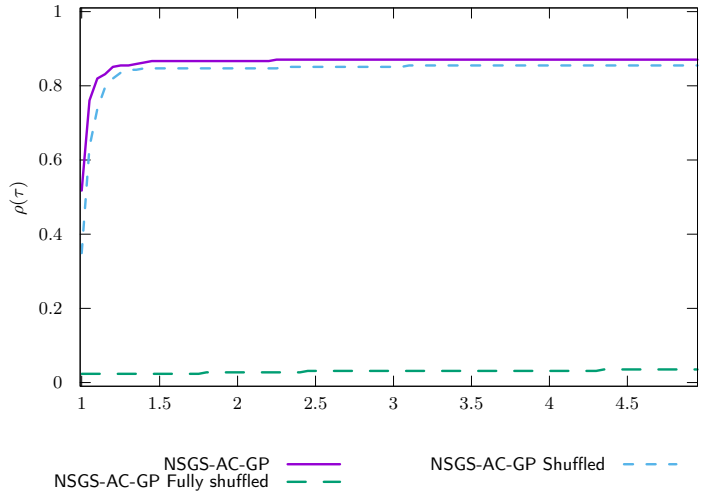


Figure 206: BoxesStack1 time NSGS/Shuffled

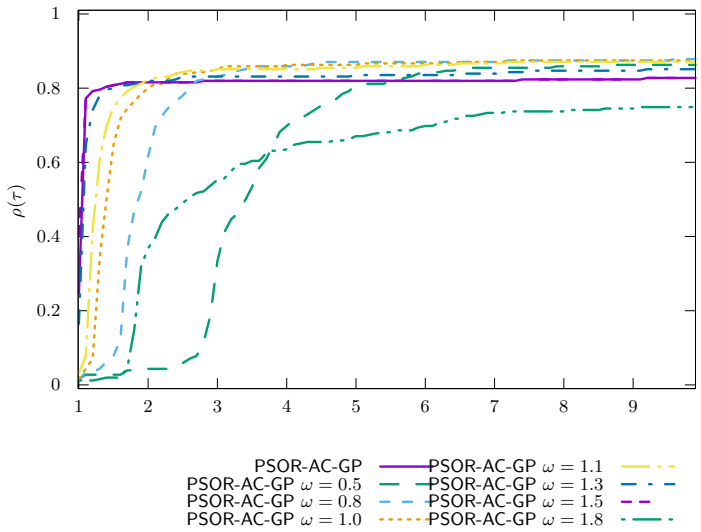


Figure 207: BoxesStack1 time PSOR

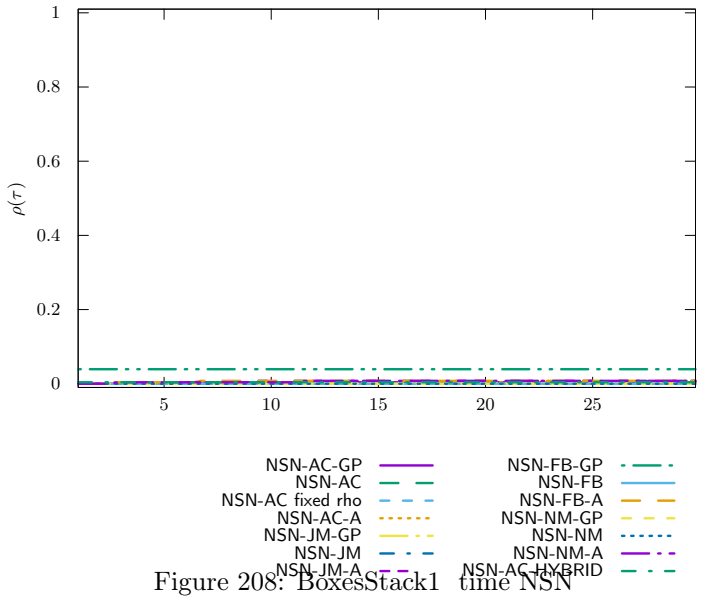


Figure 208: BoxesStack1 time NSN

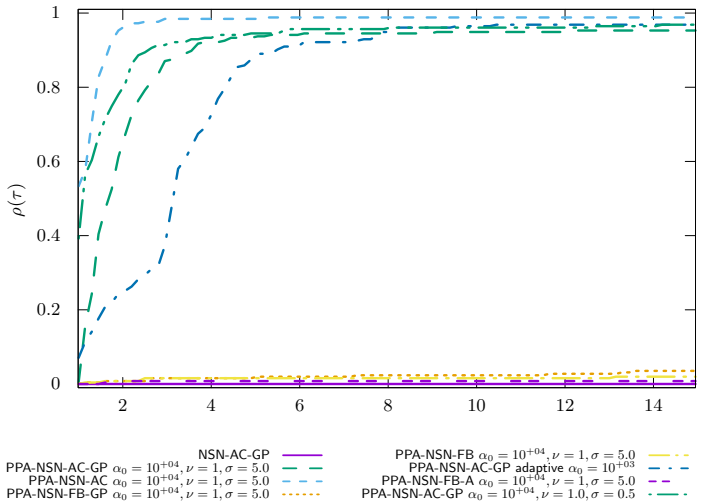


Figure 209: BoxesStack1 time PROX/NSN/InternalSolvers

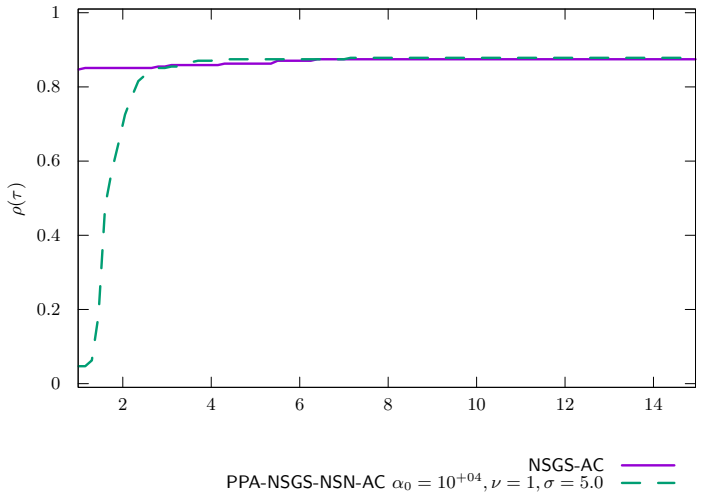


Figure 210: BoxesStack1 time PROX/NSGS/InternalSolvers

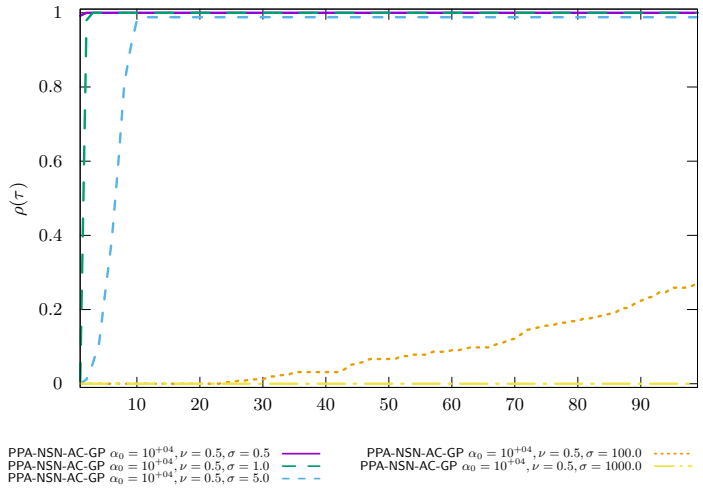


Figure 211: BoxesStack1 time PROX/Parametric studies $\nu = 0.5$

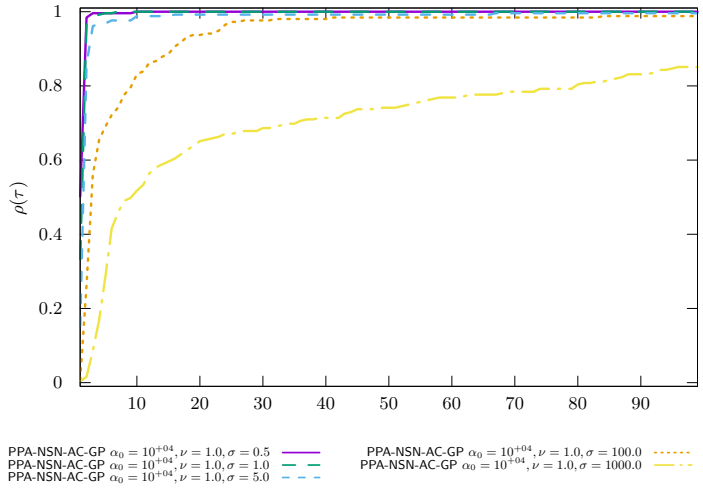


Figure 212: BoxesStack1 time PROX/Parametric studies $\nu = 1.0$

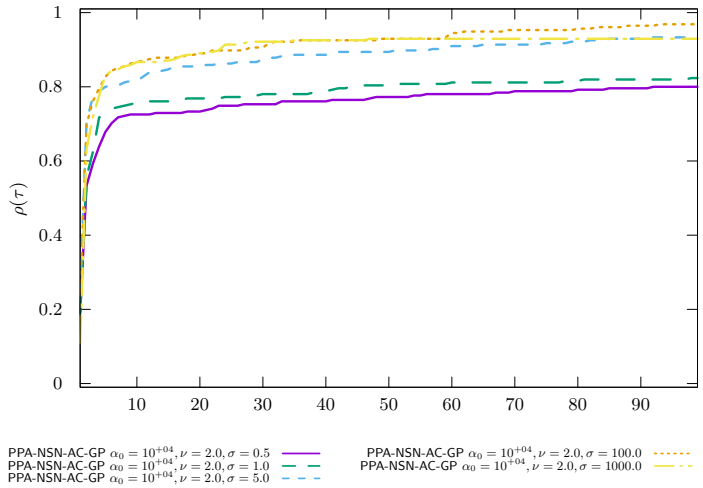


Figure 213: BoxesStack1 time PROX/Parametric studies $\nu = 2.0$

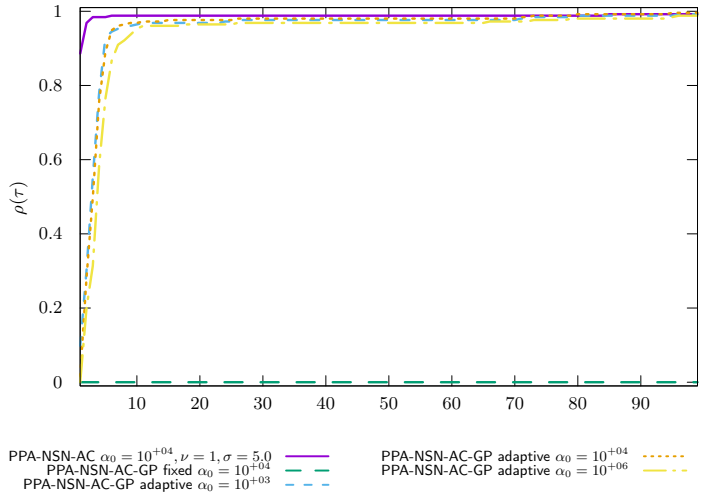


Figure 214: BoxesStack1 time PROX/Regularized problem

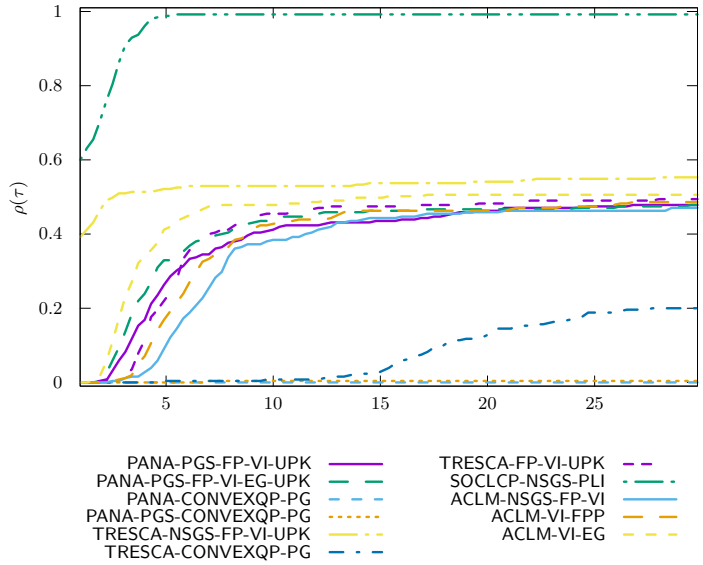


Figure 215: BoxesStack1 time OPTI

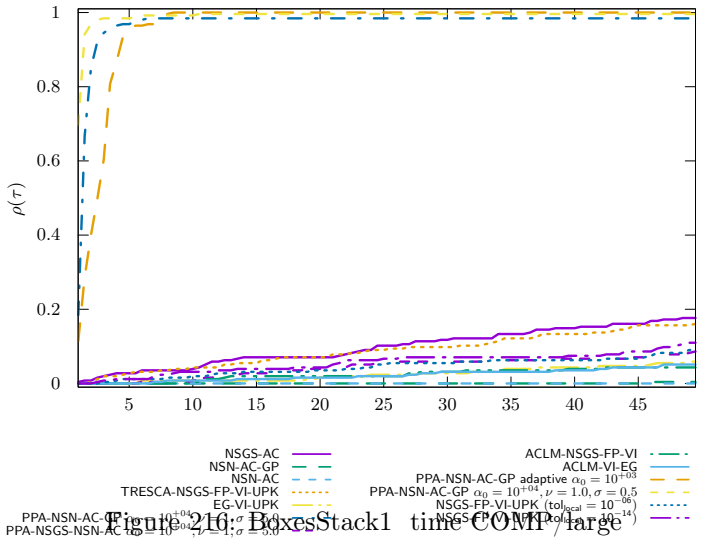


Figure 216: BoxesStack1 time COMP/large

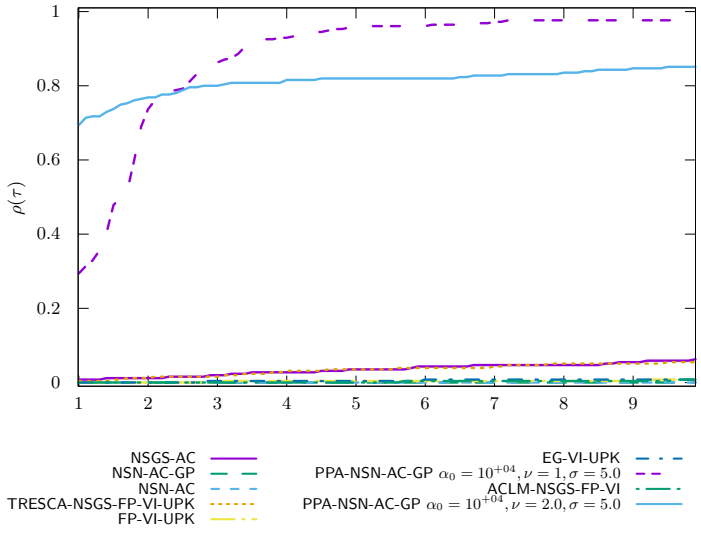


Figure 217: BoxesStack1 time COMP/zoom

KaplasTower precision 1.0e-04 timeout 100

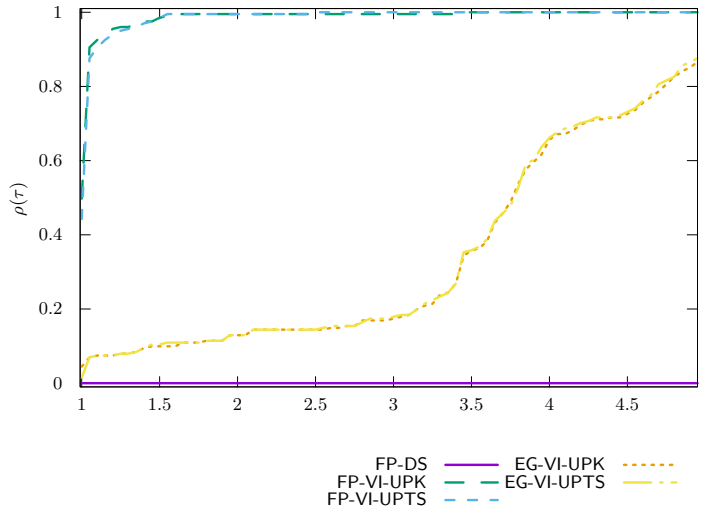


Figure 218: KaplasTower time VI/UpdateRule

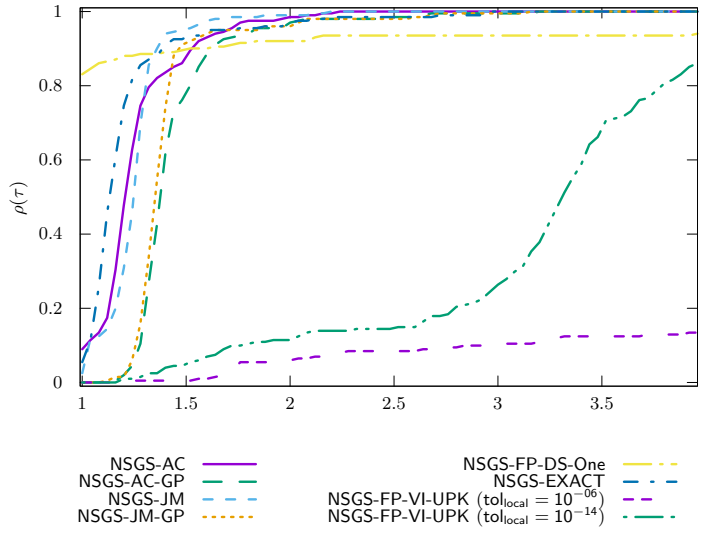


Figure 219: KaplasTower time NSGS/LocalSolver

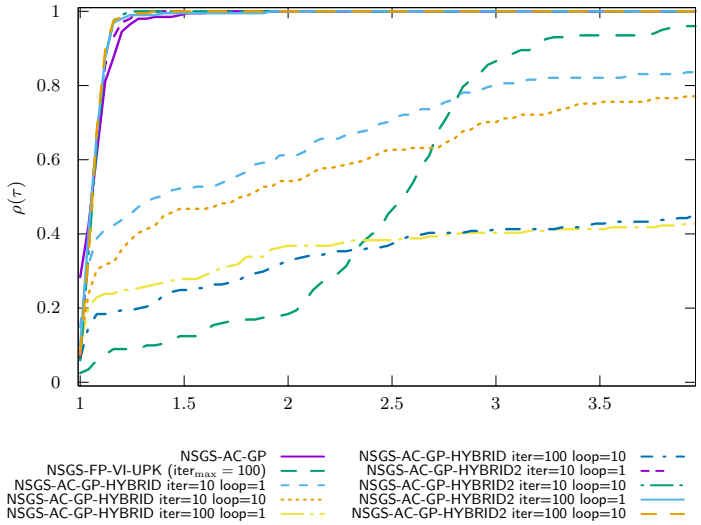


Figure 220: KaplasTower time NSGS/LocalSolverHybrid

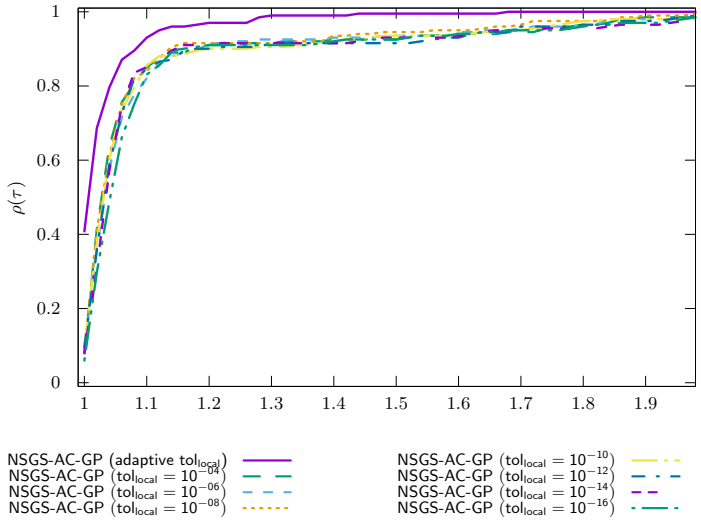


Figure 221: KaplasTower time NSGS/LocalTol

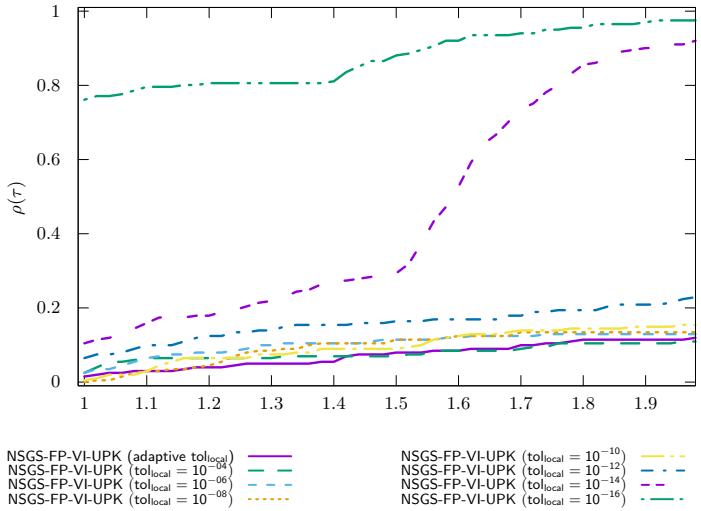


Figure 222: KaplasTower time NSGS/LocalTol-VI

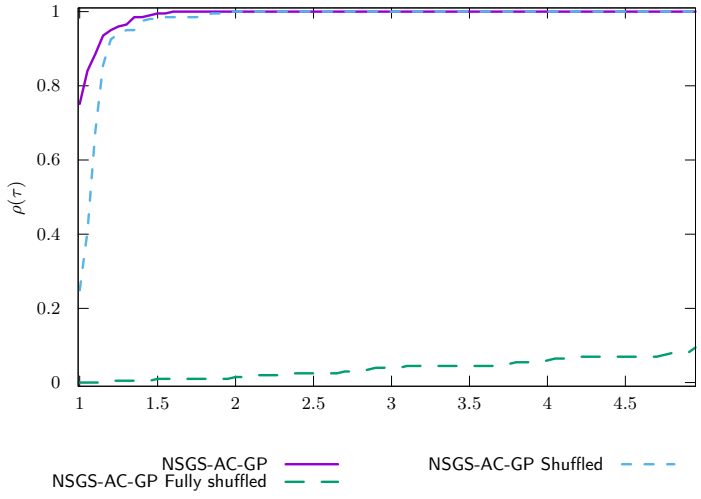


Figure 223: KaplasTower time NSGS/Shuffled

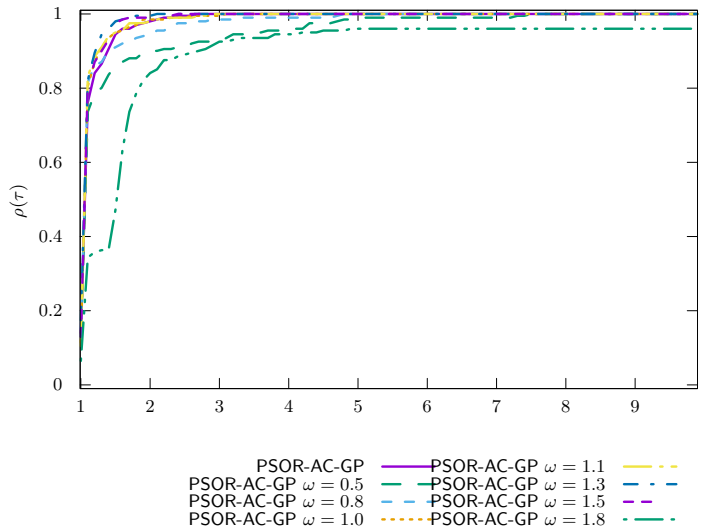


Figure 224: KaplasTower time PSOR

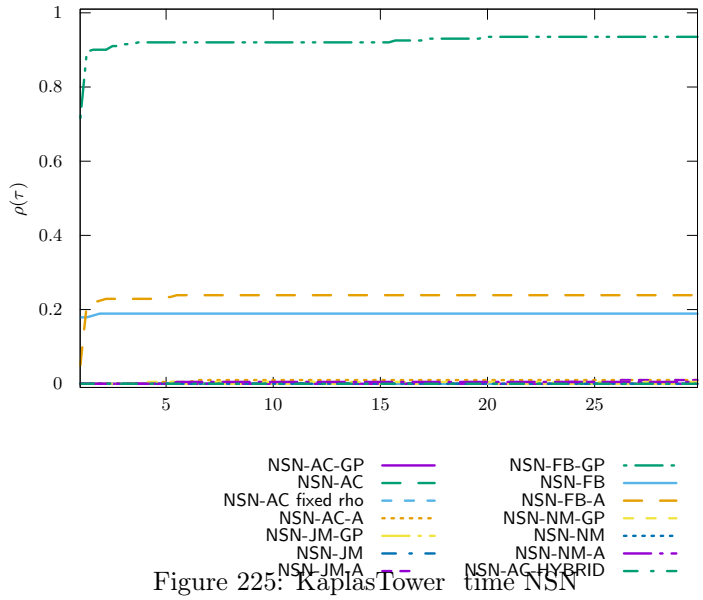


Figure 225: KaplasTower time NSN

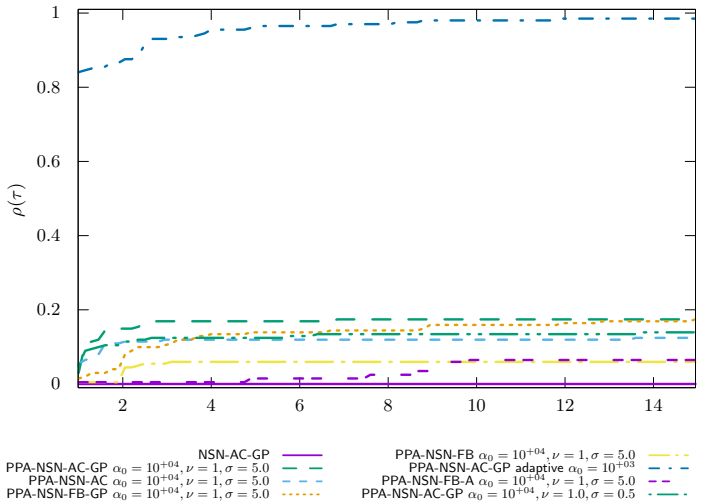


Figure 226: KaplasTower time PROX/NSN/InternalSolvers

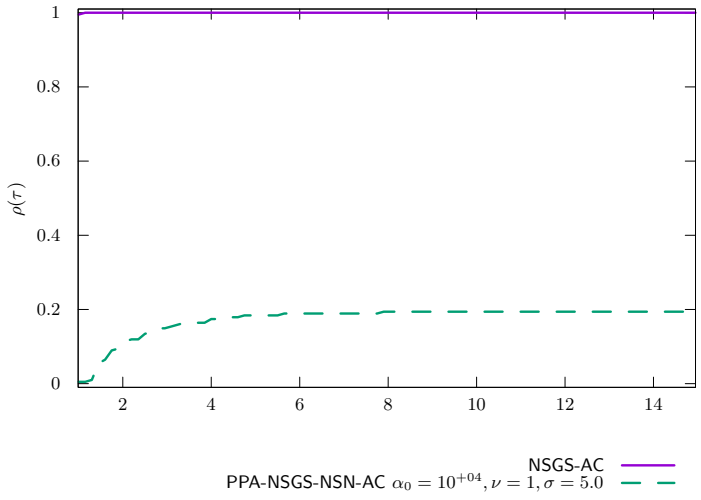


Figure 227: KaplasTower time PROX/NSGS/InternalSolvers

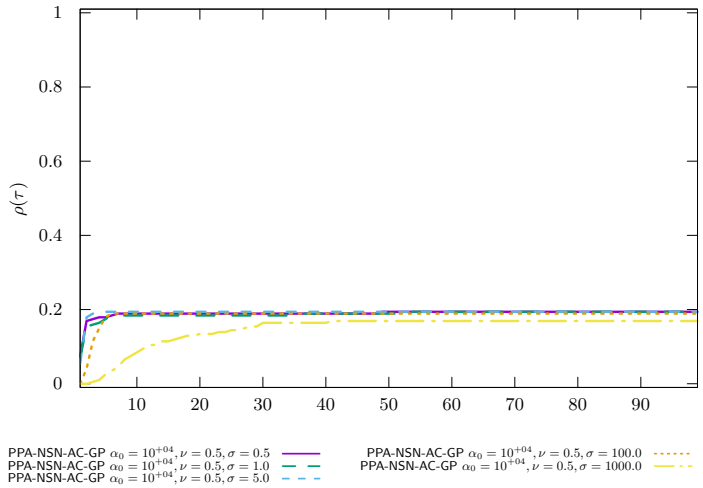


Figure 228: KaplasTower time PROX/Parametric studies $\nu = 0.5$

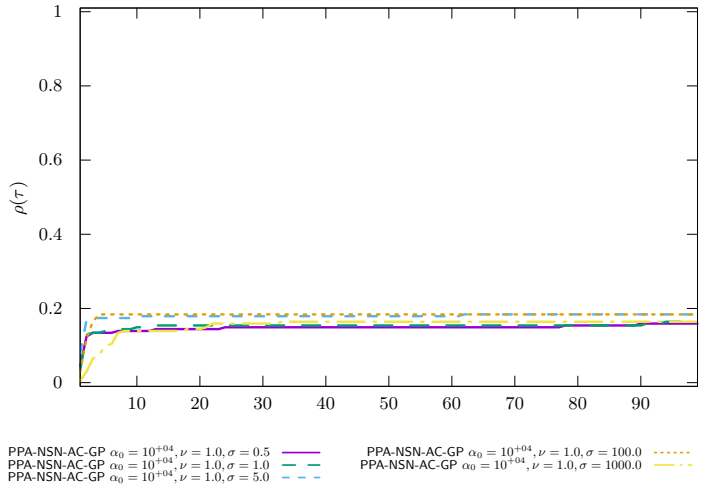


Figure 229: KaplasTower time PROX/Parametric studies $\nu = 1.0$

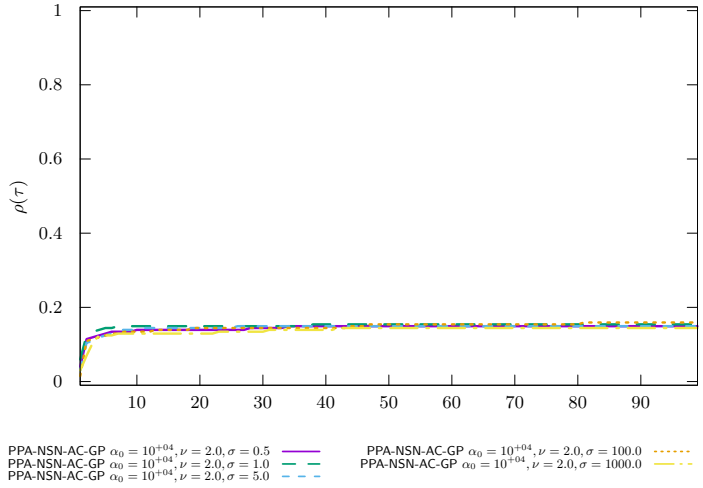


Figure 230: KaplasTower time PROX/Parametric studies $\nu = 2.0$

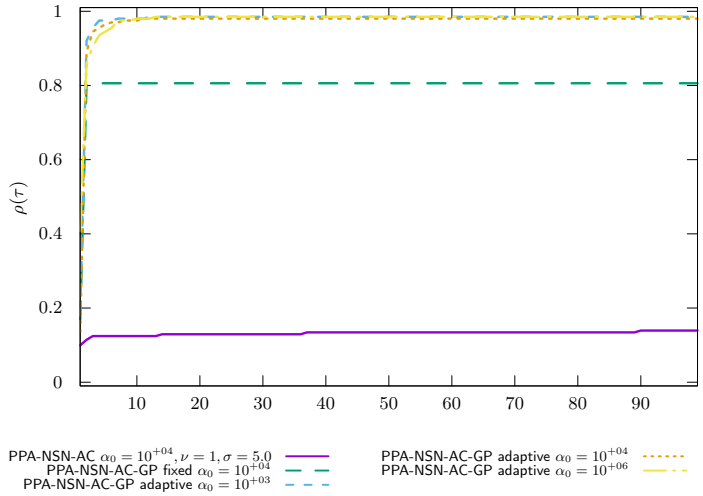


Figure 231: KaplasTower time PROX/Regularized problem

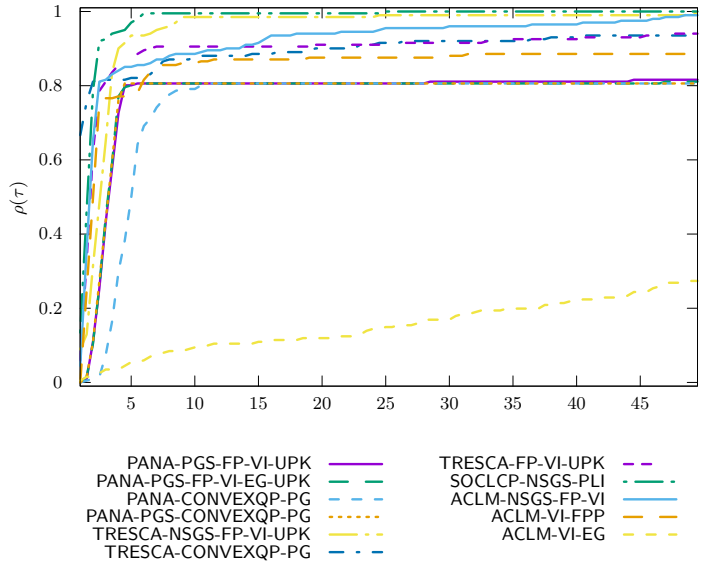


Figure 232: KaplasTower time OPTI

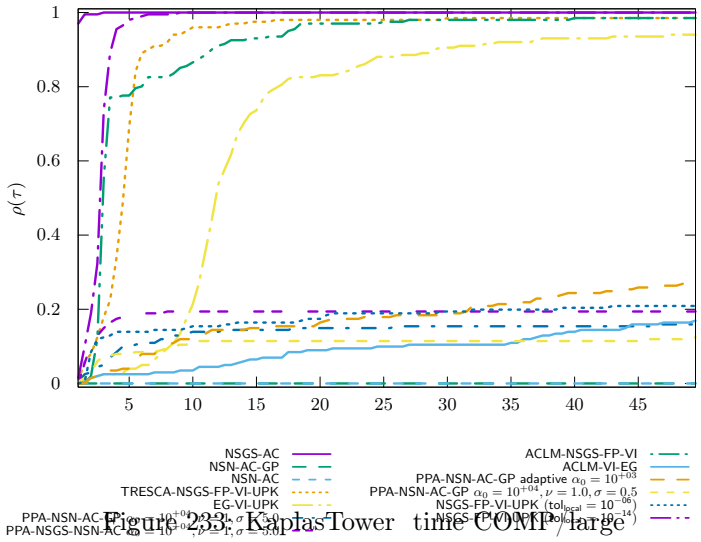


Figure 233: KaplasTower time COMP/large

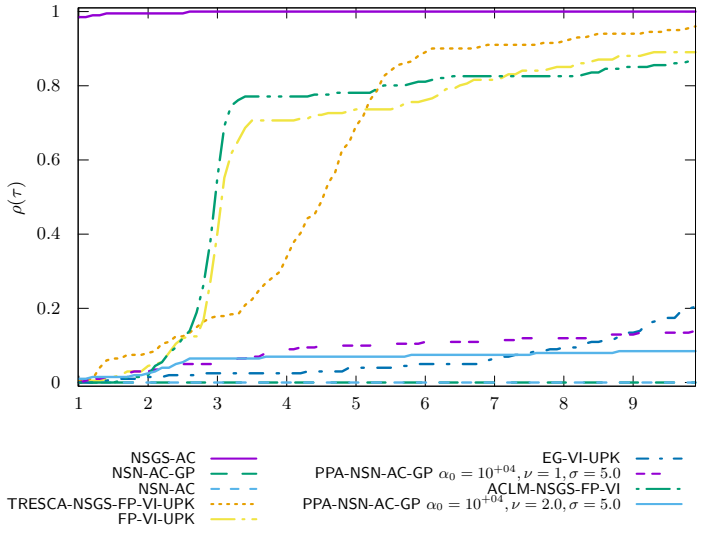


Figure 234: KaplasTower time COMP/zoom

Chute_1000 precision 1.0e-04 timeout 200

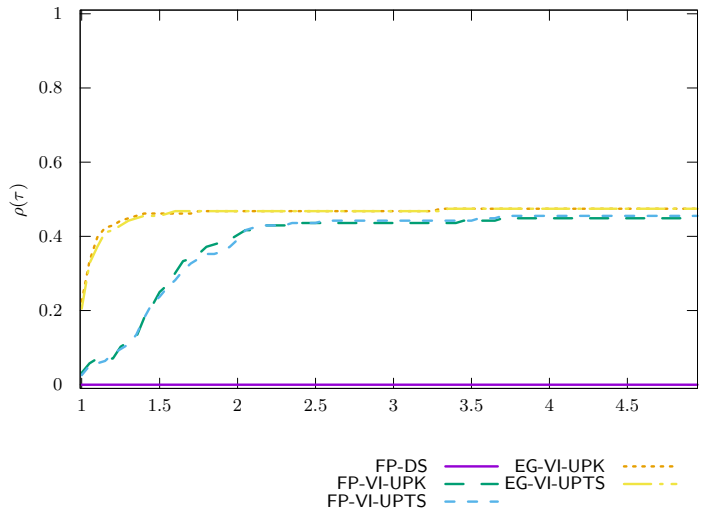


Figure 235: Chute_1000 time VI/UpdateRule

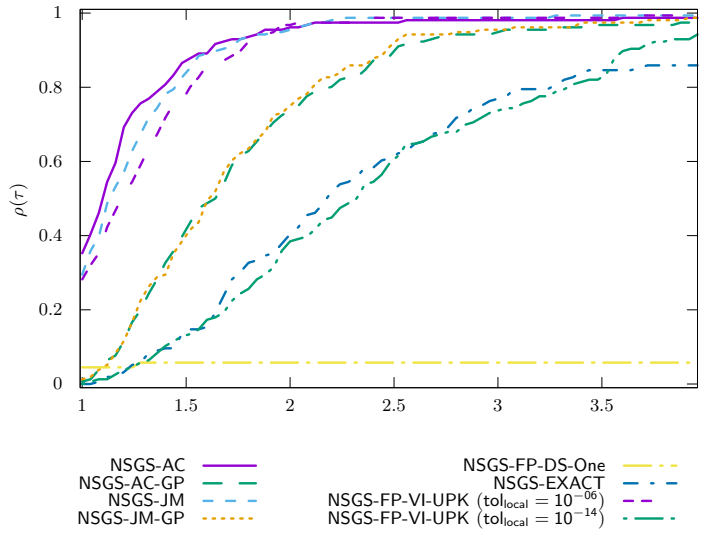


Figure 236: Chute_1000 time NSGS/LocalSolver

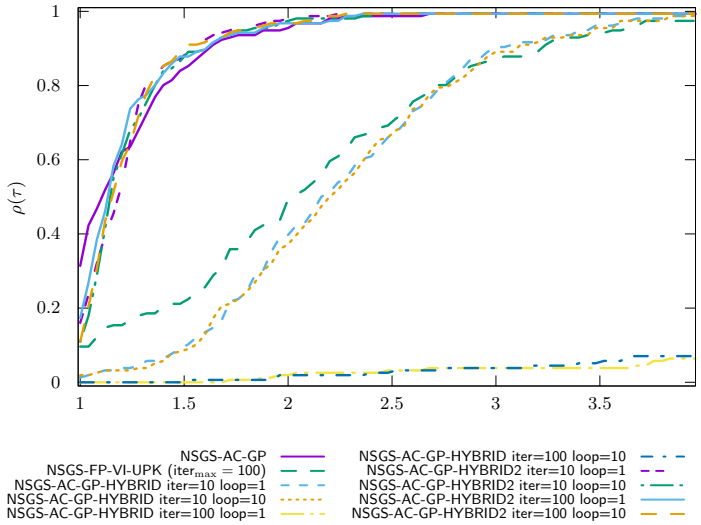


Figure 237: Chute_1000 time NSGS/LocalSolverHybrid

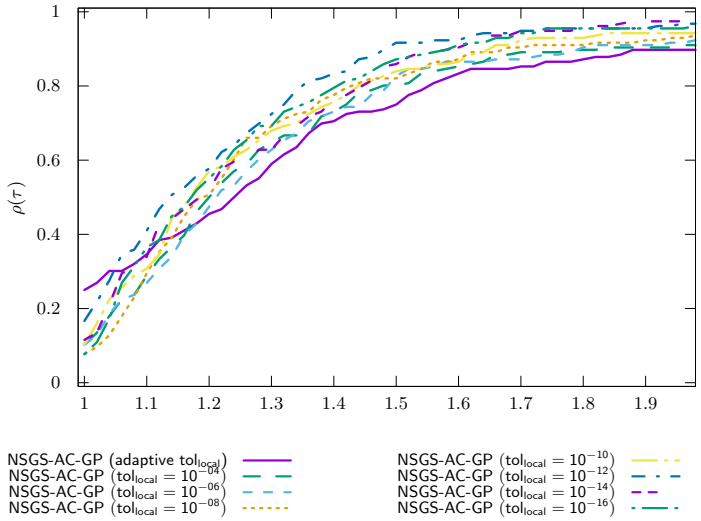


Figure 238: Chute_1000 time NSGS/LocalTol

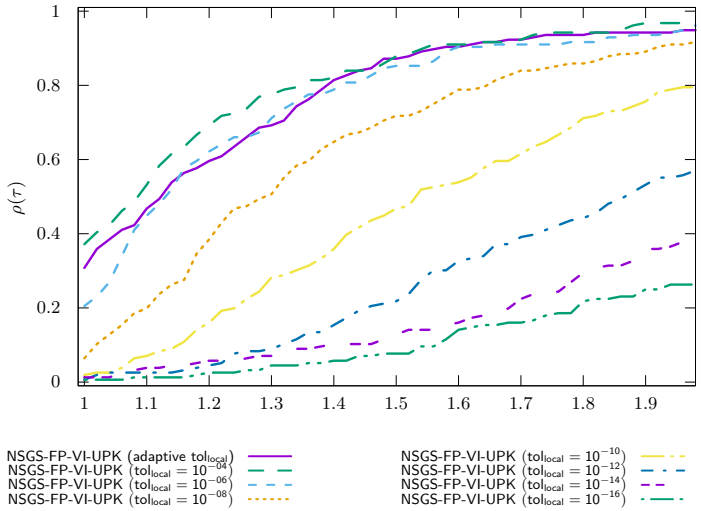


Figure 239: Chute_1000 time NSGS/LocalTol-VI

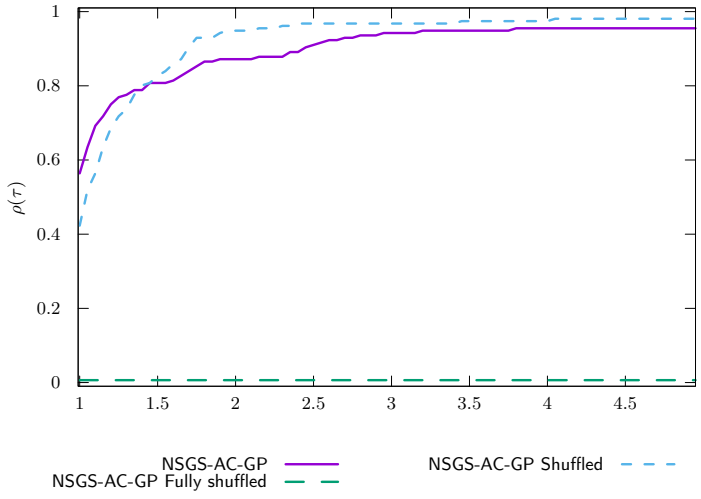


Figure 240: Chute_1000 time NSGS/Shuffled

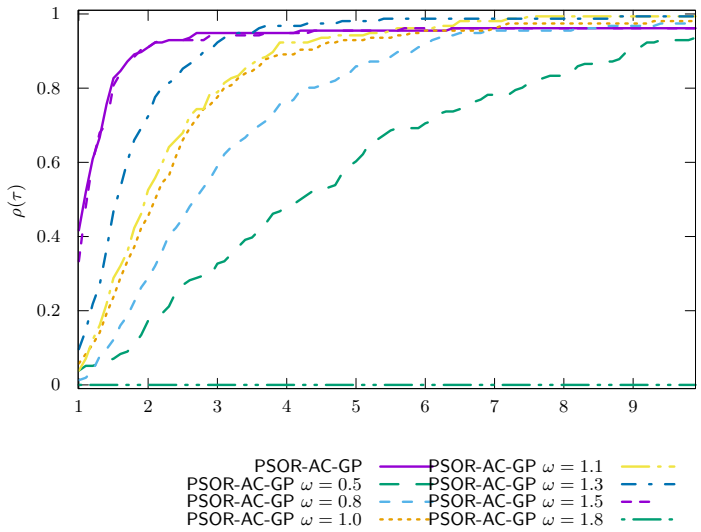


Figure 241: Chute_1000 time PSOR

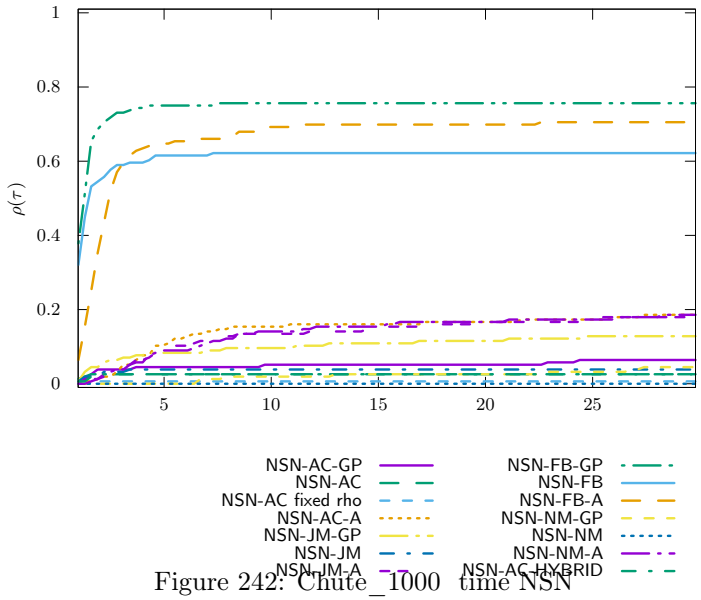


Figure 242: Chute_1000 time NSN

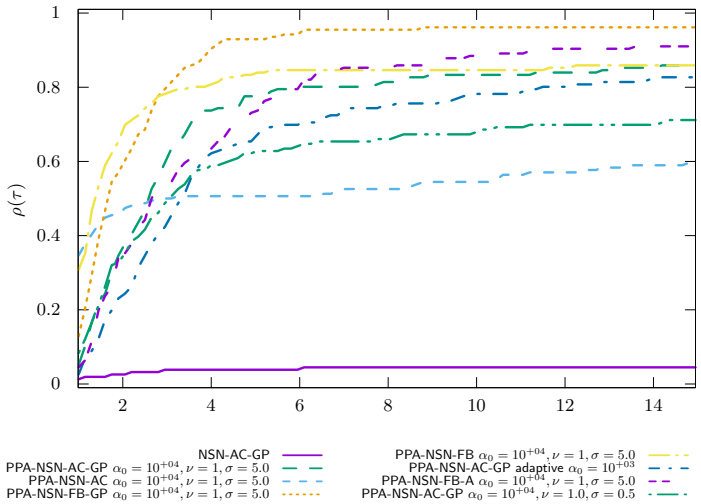


Figure 243: Chute_1000 time PROX/NSN/InternalSolvers

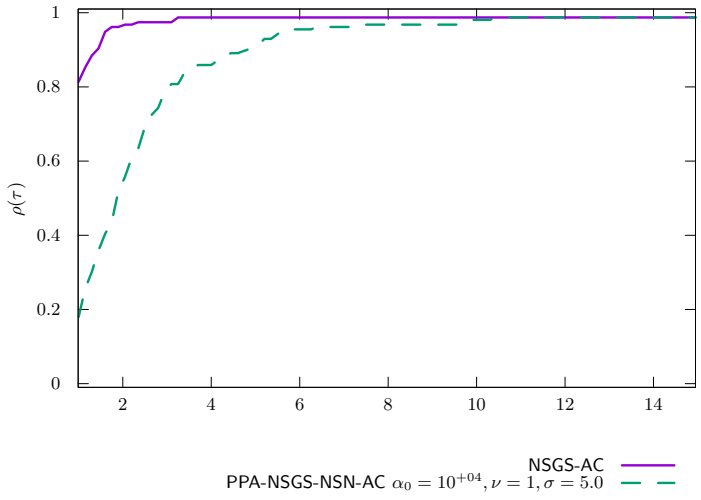


Figure 244: Chute_1000 time PROX/NSGS/InternalSolvers

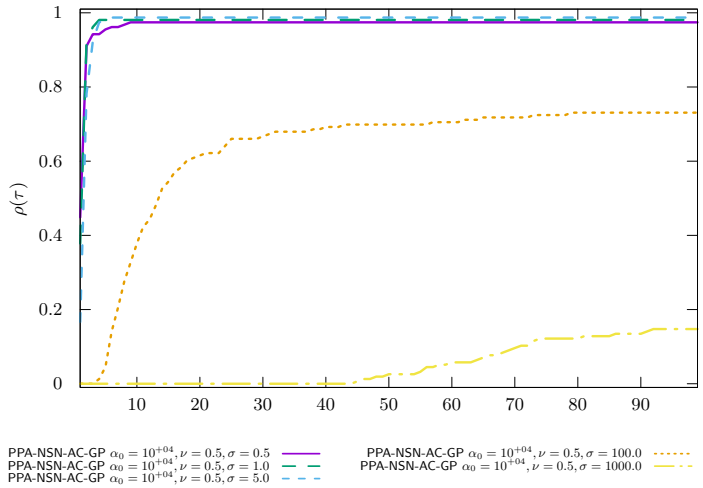


Figure 245: Chute_1000 time PROX/Parametric studies $\nu = 0.5$

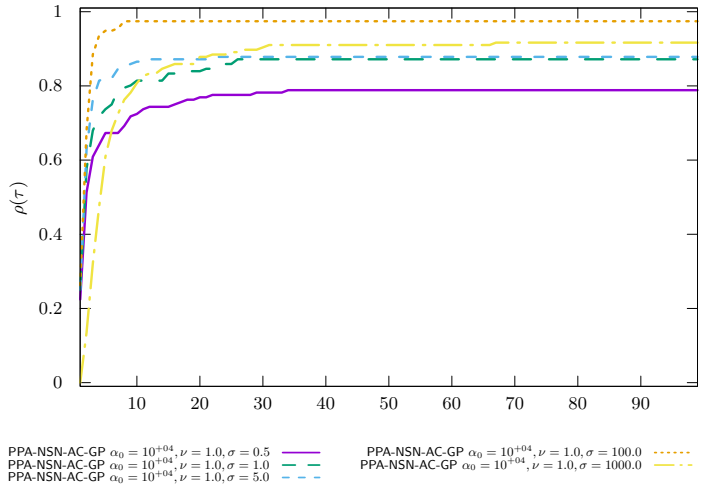


Figure 246: Chute_1000 time PROX/Parametric studies $\nu = 1.0$

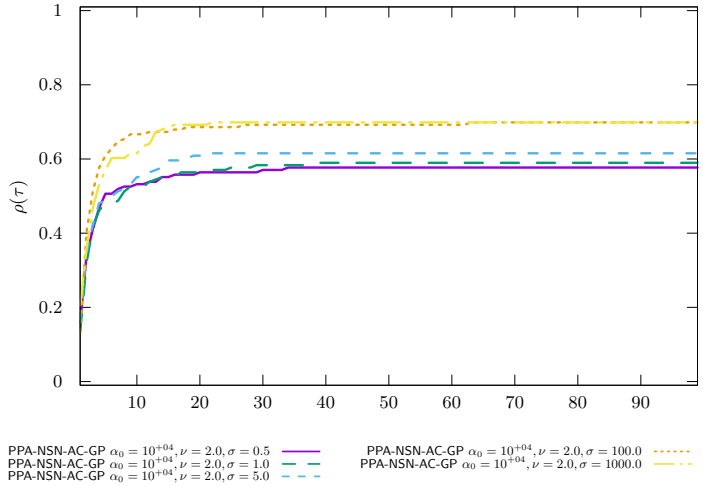


Figure 247: Chute_1000 time PROX/Parametric studies $\nu = 2.0$

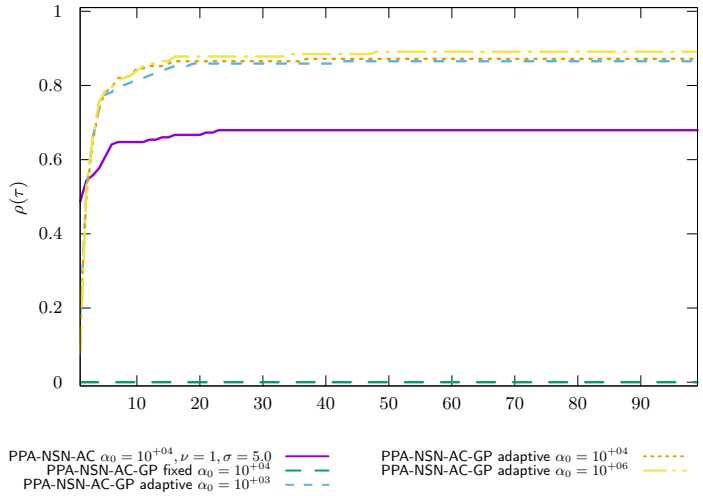


Figure 248: Chute_1000 time PROX/Regularized problem

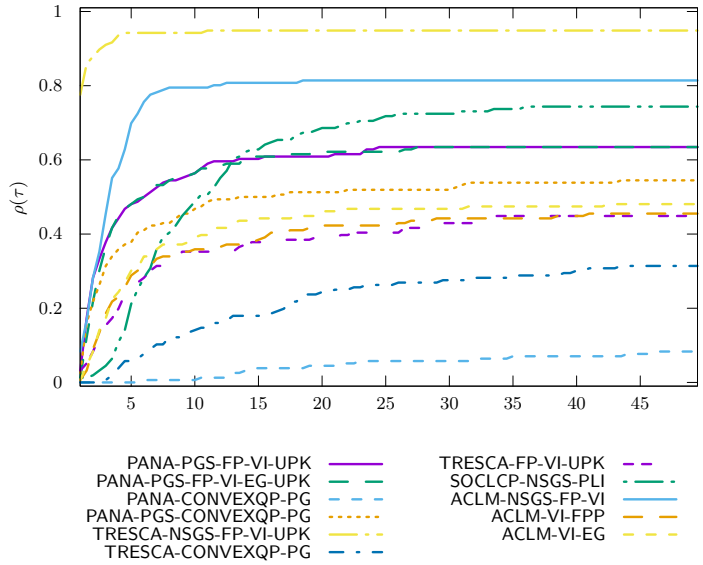


Figure 249: Chute_1000 time OPTI

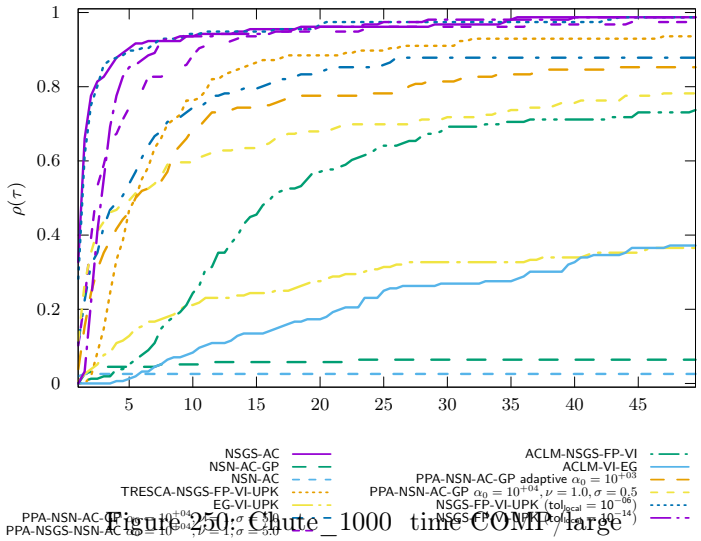


Figure 250: Chute_1000 time COMP/large

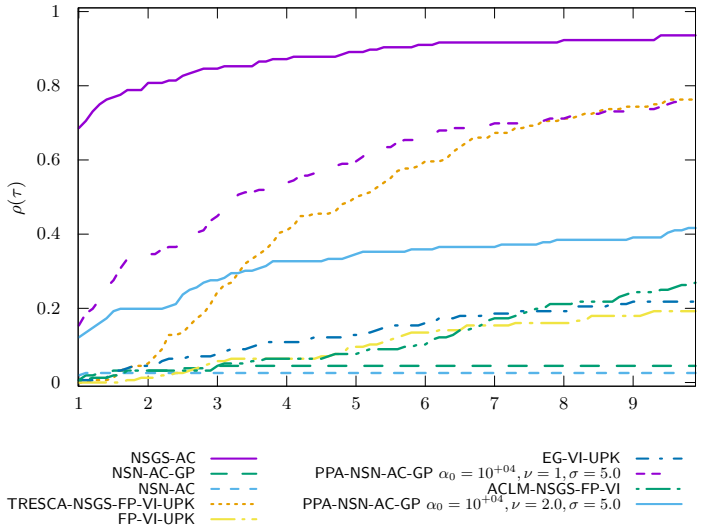


Figure 251: Chute_1000 time COMP/zoom

Chute_4000 precision 1.0e-04 timeout 200

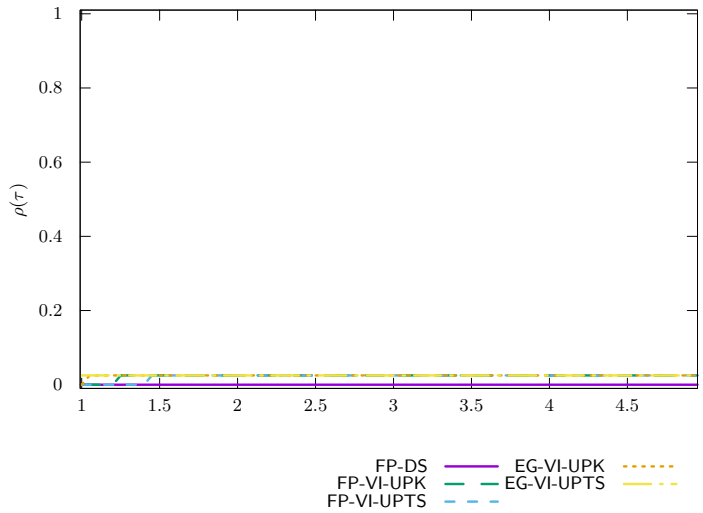


Figure 252: Chute_4000 time VI/UpdateRule

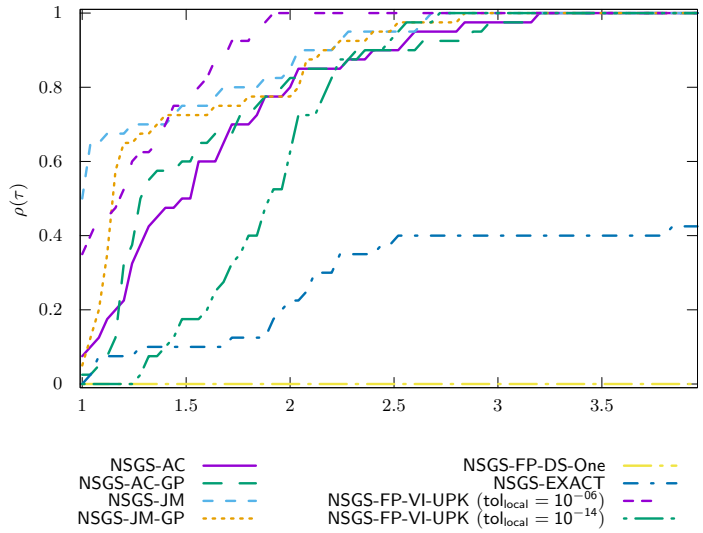


Figure 253: Chute_4000 time NSGS/LocalSolver

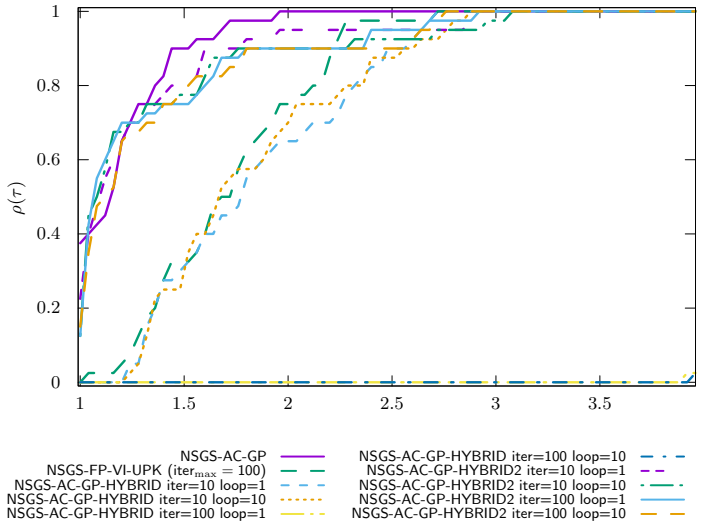


Figure 254: Chute_4000 time NSGS/LocalSolverHybrid

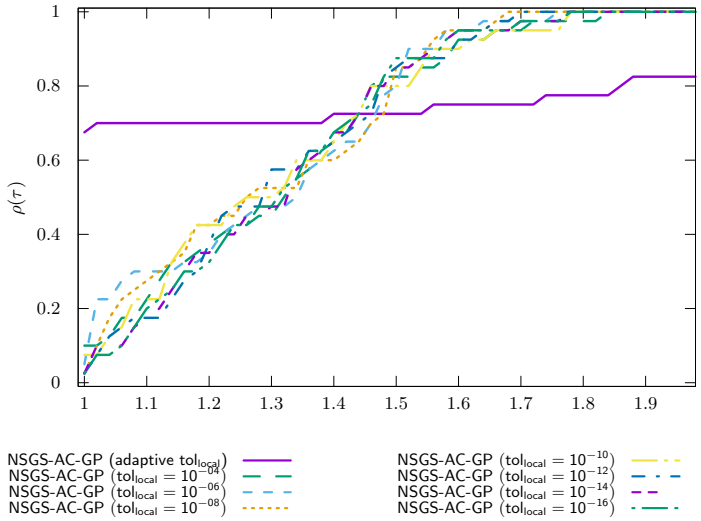


Figure 255: Chute_4000 time NSGS/LocalTol

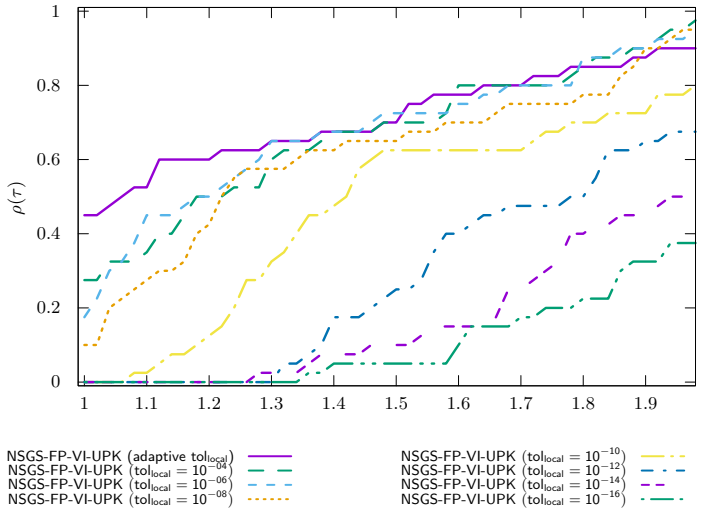


Figure 256: Chute_4000 time NSGS/LocalTol-VI

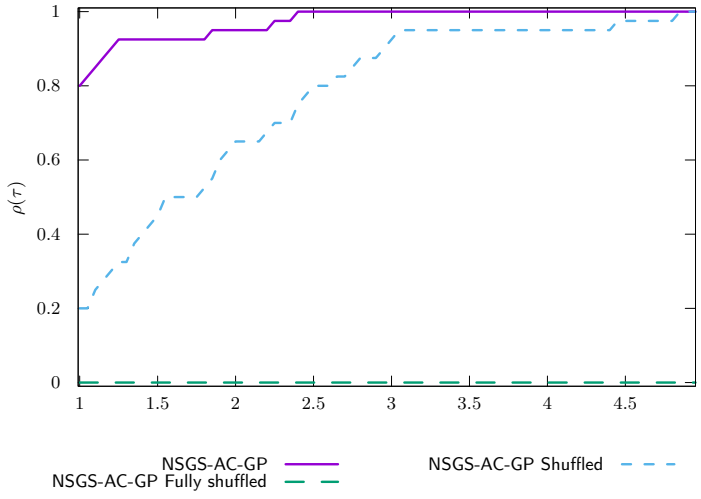


Figure 257: Chute_4000 time NSGS/Shuffled

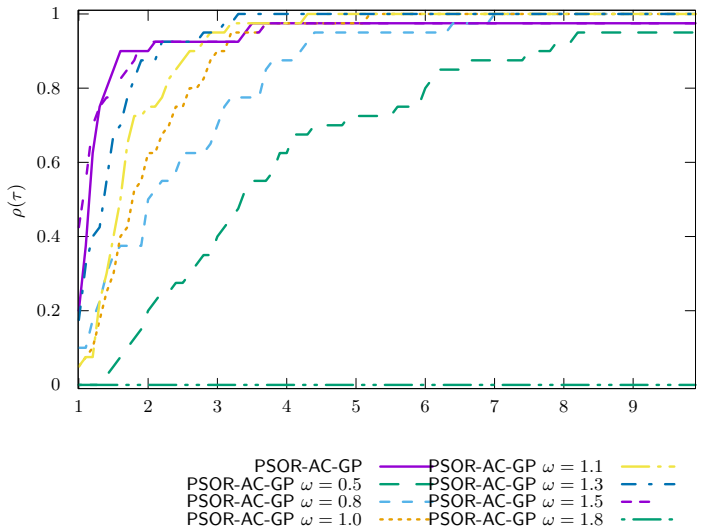


Figure 258: Chute_4000 time PSOR

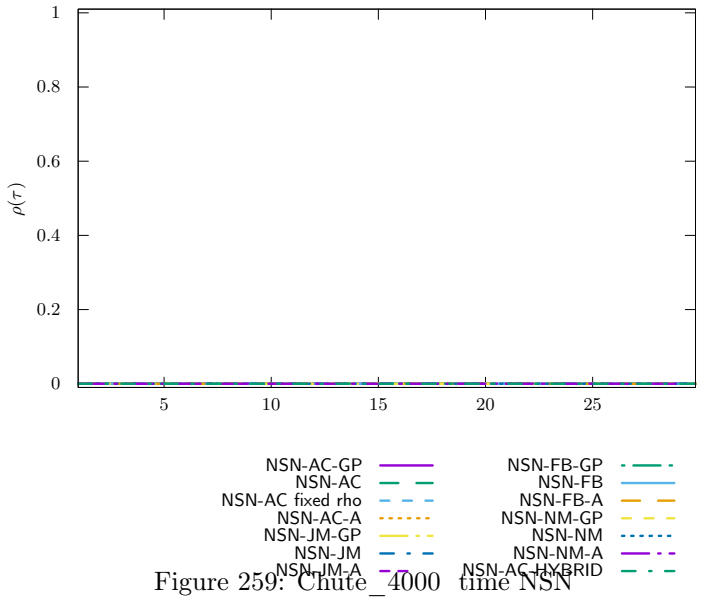


Figure 259: Chute_4000 time NSN

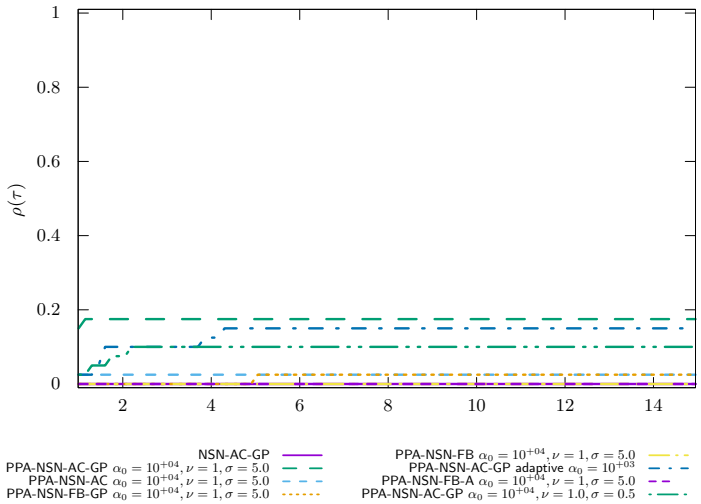


Figure 260: Chute_4000 time PROX/NSN/InternalSolvers

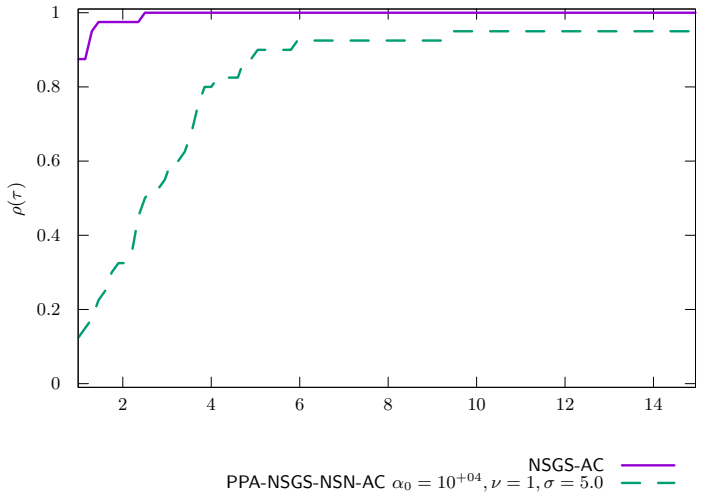


Figure 261: Chute_4000 time PROX/NSGS/InternalSolvers

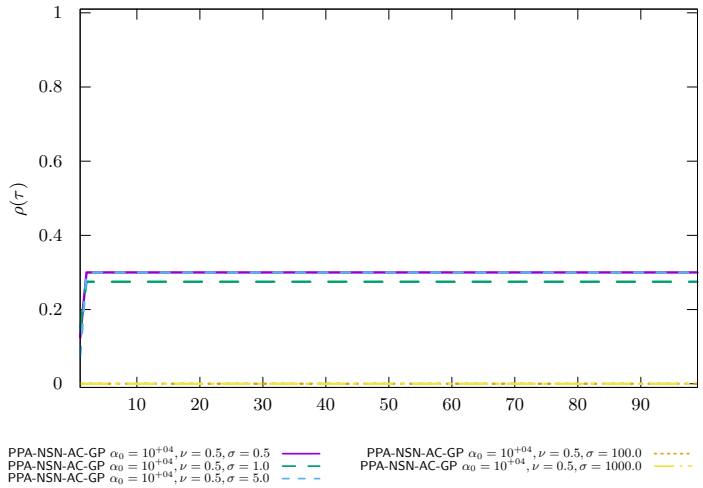


Figure 262: Chute_4000 time PROX/Parametric studies $\nu = 0.5$

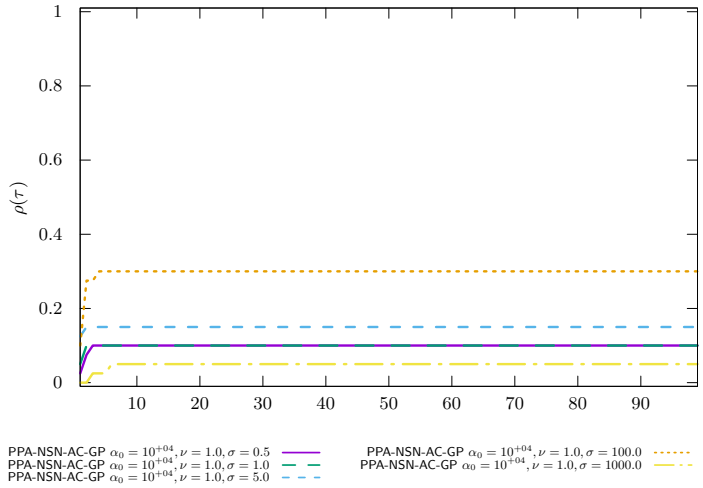


Figure 263: Chute_4000 time PROX/Parametric studies $\nu = 1.0$

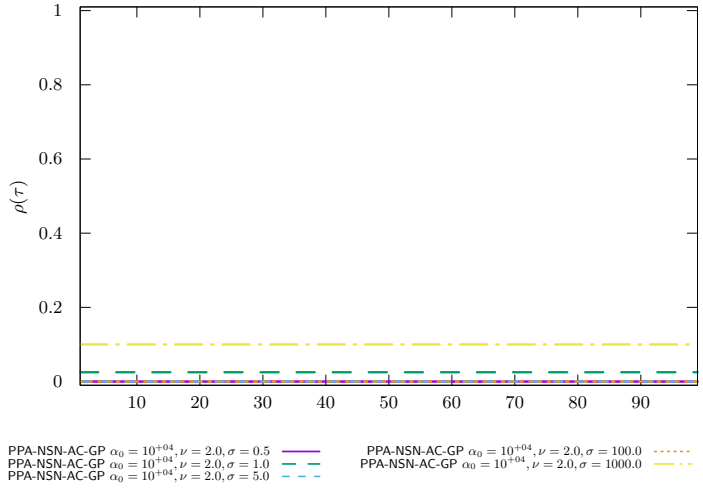


Figure 264: Chute_4000 time PROX/Parametric studies $\nu = 2.0$

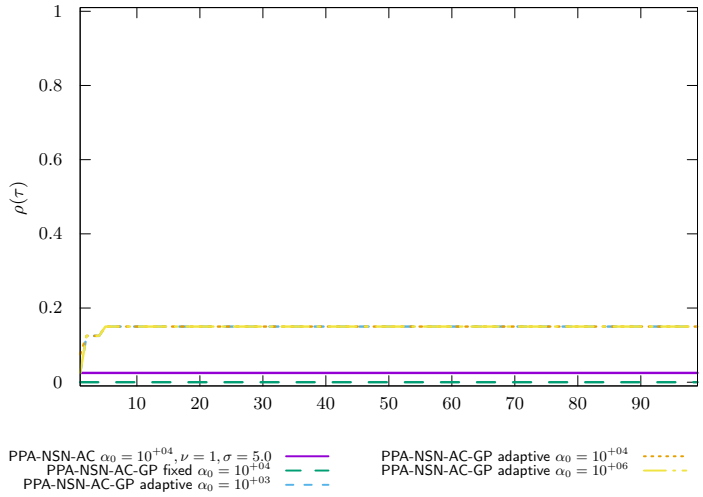


Figure 265: Chute_4000 time PROX/Regularized problem

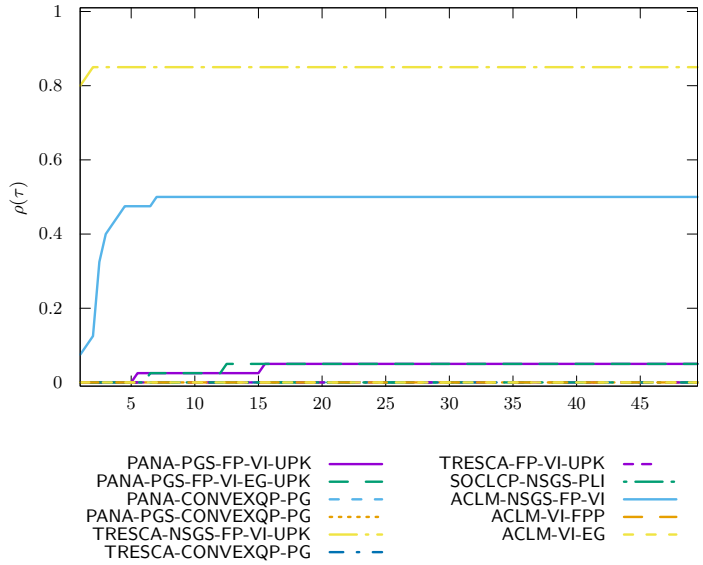


Figure 266: Chute_4000 time OPTI

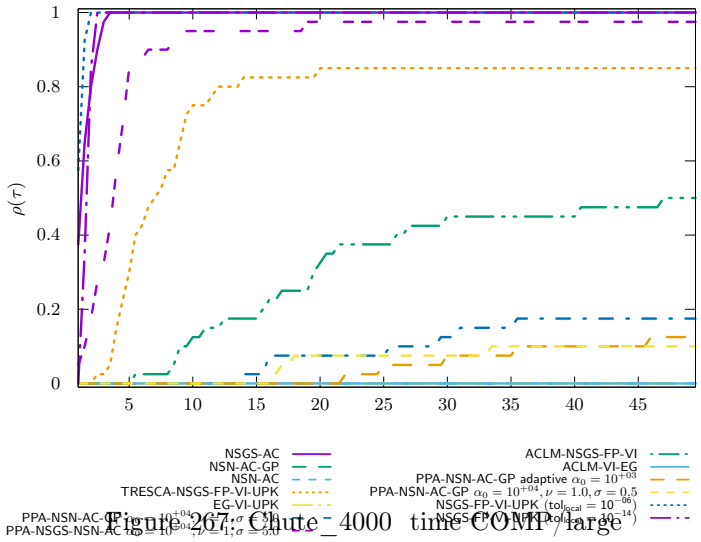


Figure 267: Chute_4000 time COMP/large

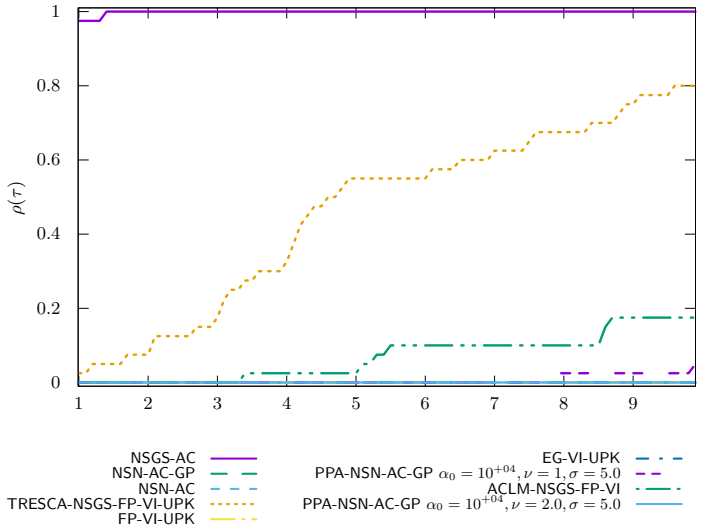


Figure 268: Chute_4000 time COMP/zoom

Chute_local_problems precision 1.0e-04 timeout 10

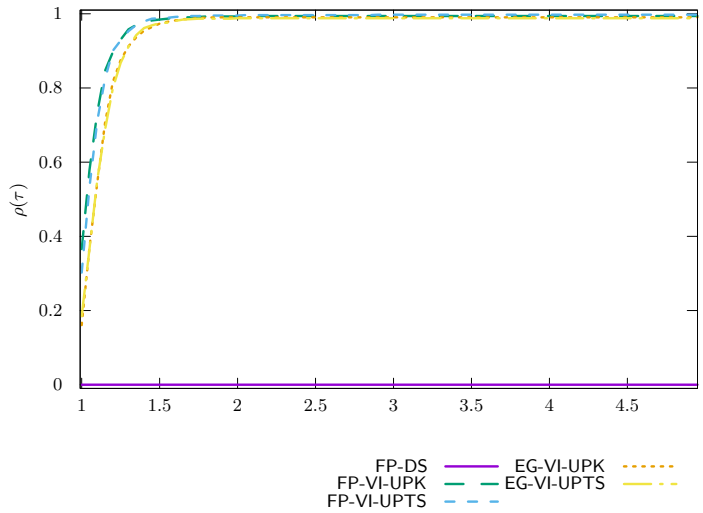


Figure 269: Chute_local_problems time VI/UpdateRule

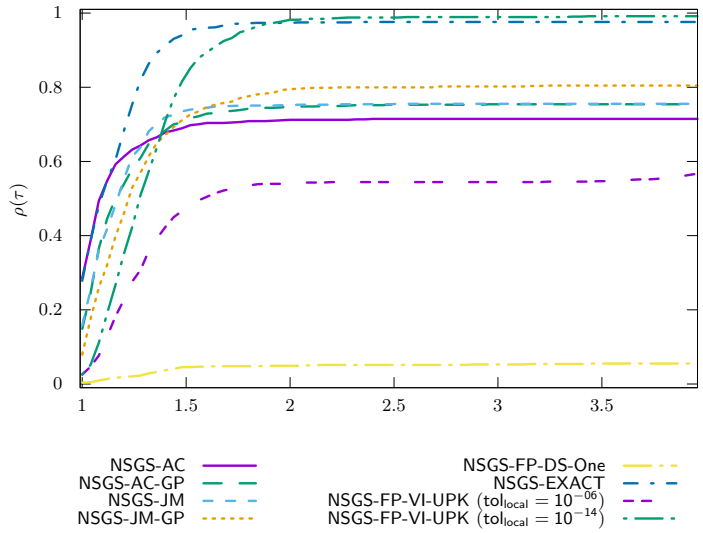


Figure 270: Chute_local_problems time NSGS/LocalSolver

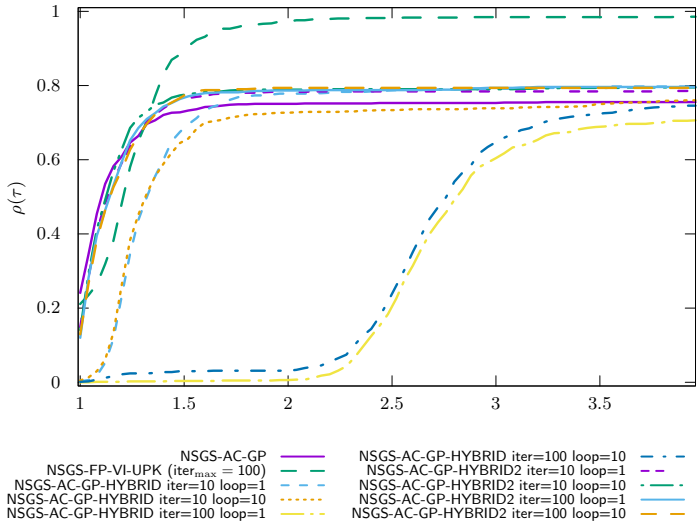


Figure 271: Chute_local_problems time NSGS/LocalSolverHybrid

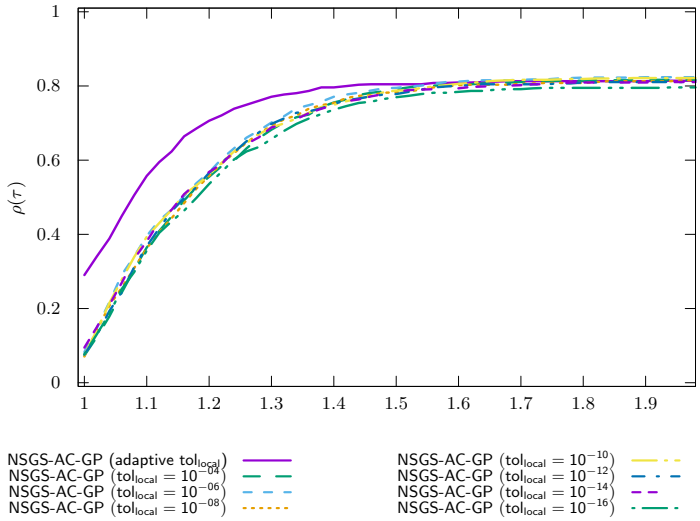


Figure 272: Chute_local_problems time NSGS/LocalTol

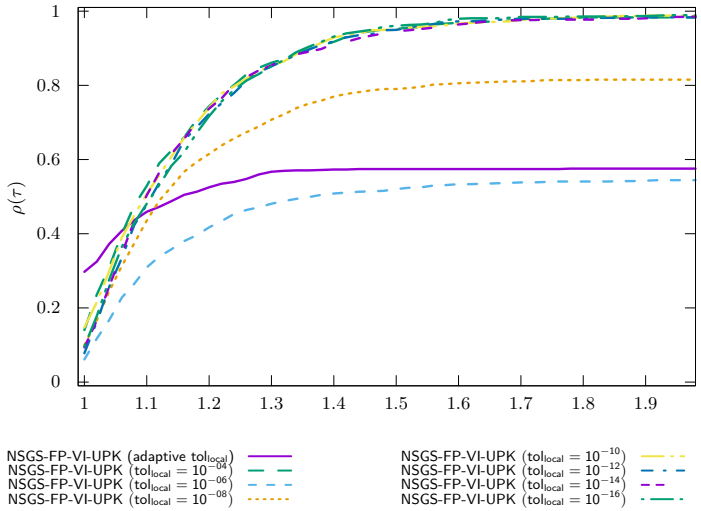


Figure 273: Chute_local_problems time NSGS/LocalTol-VI

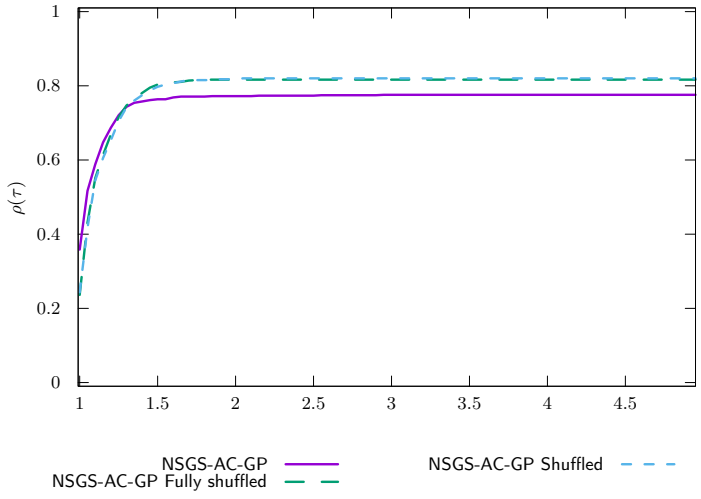


Figure 274: Chute_local_problems time NSGS/Shuffled

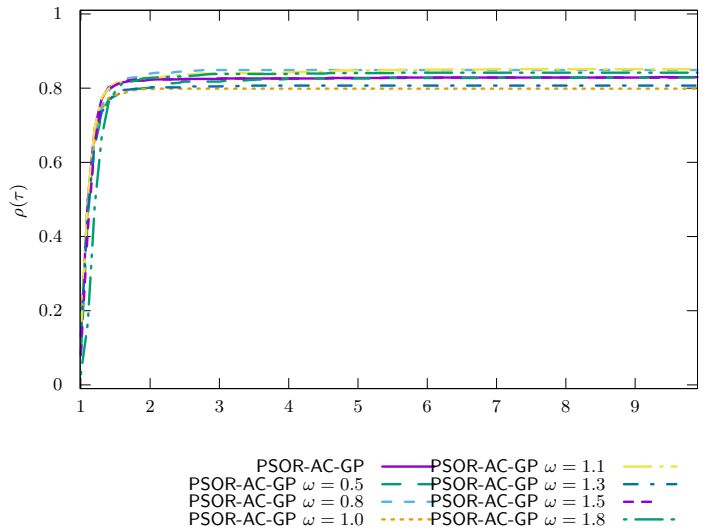


Figure 275: Chute_local_problems time PSOR

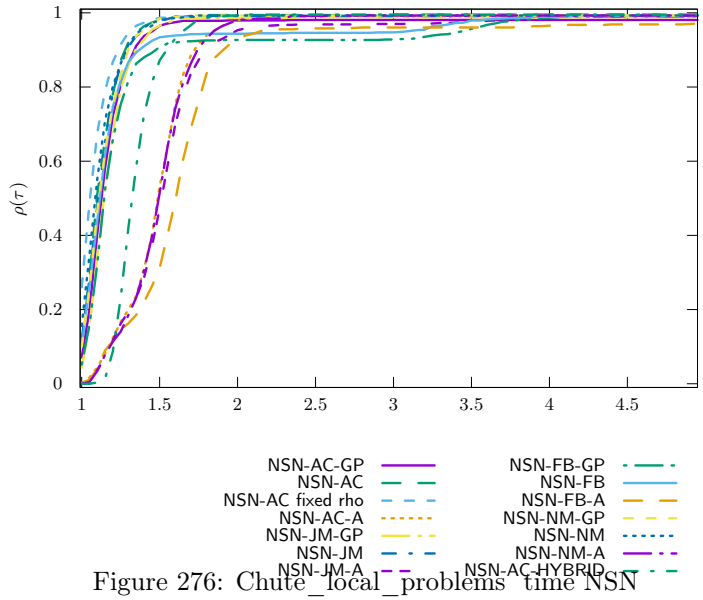


Figure 276: Chute_local_problems time NSN

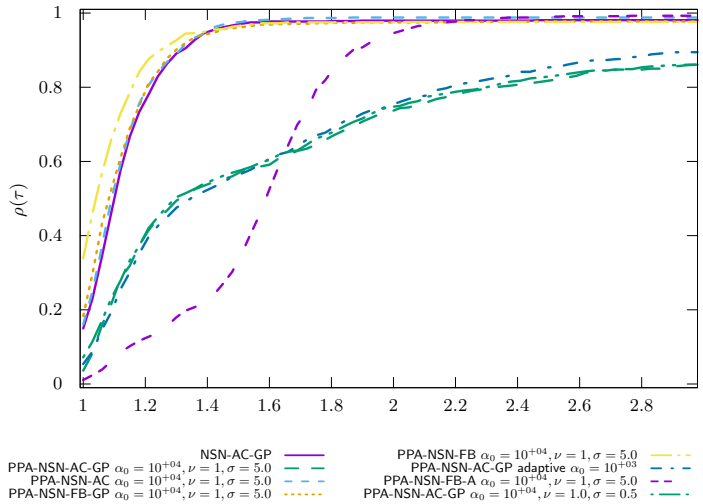


Figure 277: Chute_local_problems time PROX/NSN/InternalSolvers

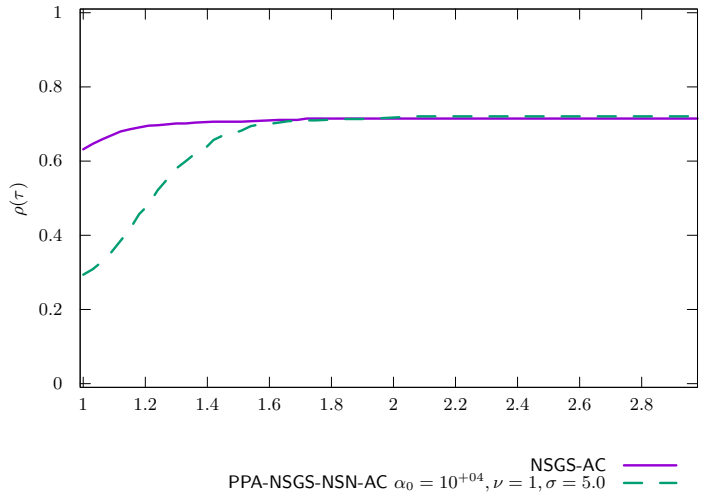


Figure 278: Chute_local_problems time PROX/NSGS/InternalSolvers

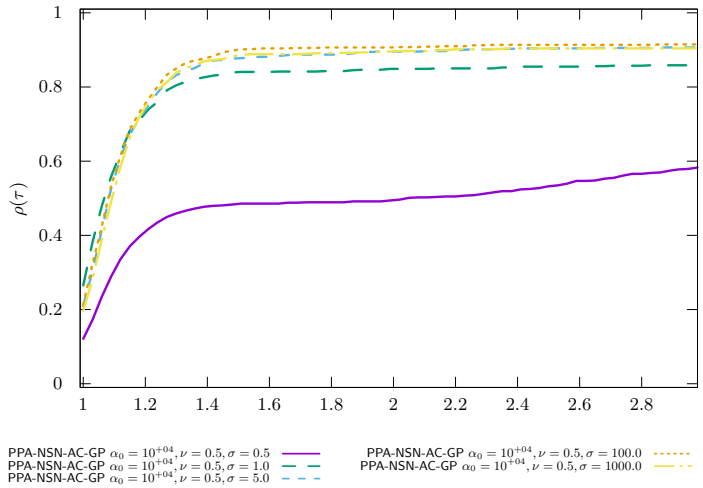


Figure 279: Chute_local_problems time PROX/Parametric studies $\nu = 0.5$

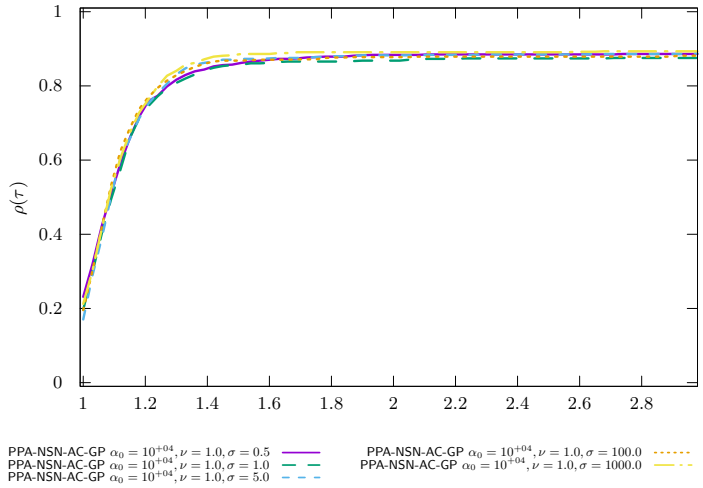


Figure 280: Chute_local_problems time PROX/Parametric studies $\nu = 1.0$

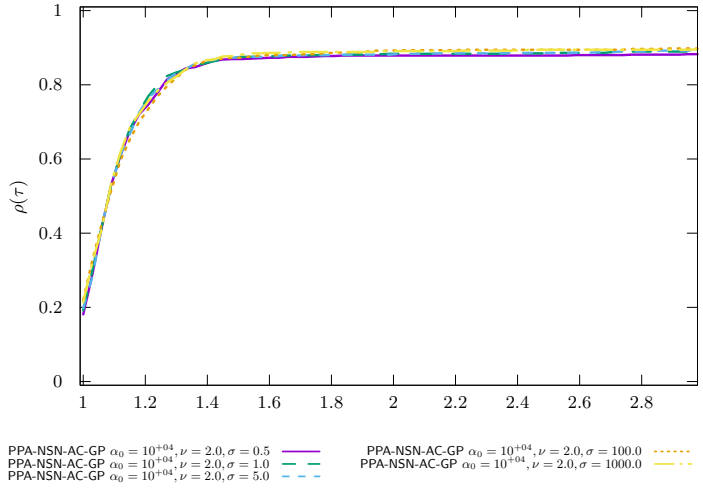


Figure 281: Chute_local_problems time PROX/Parametric studies $\nu = 2.0$

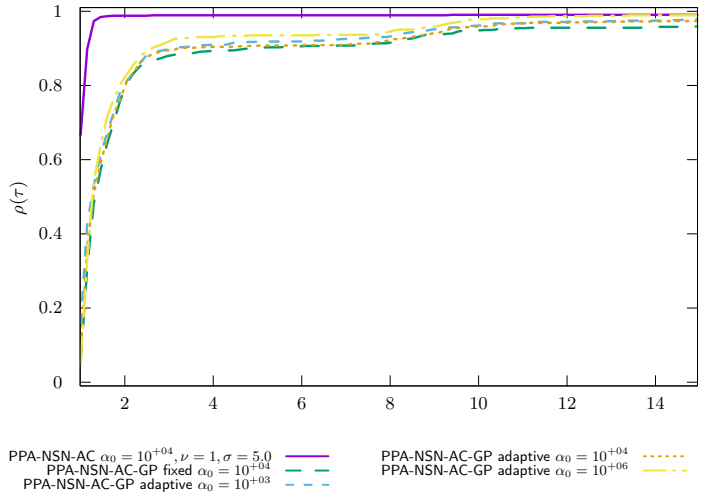


Figure 282: Chute_local_problems time PROX/Regularized problem

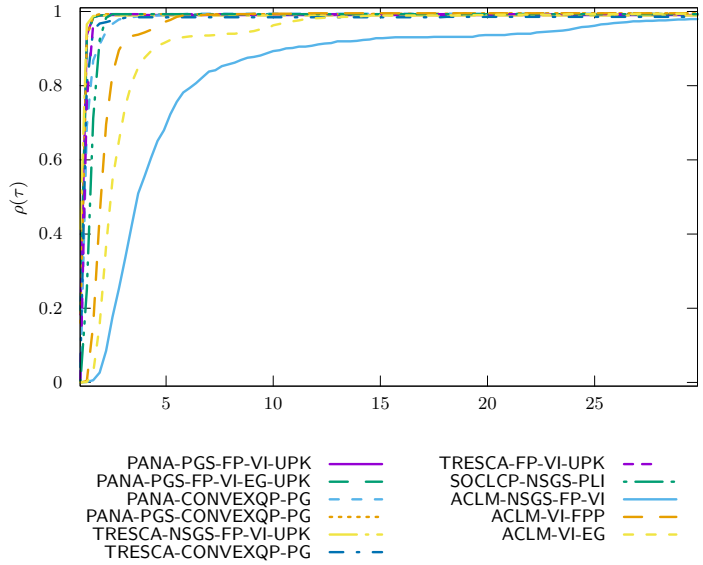


Figure 283: Chute_local_problems time OPTI

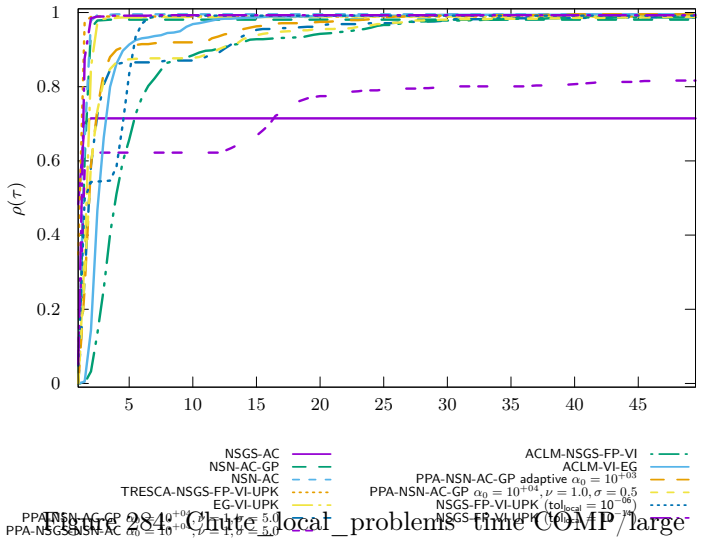


Figure 284: Chute_local_problems time COMP/large

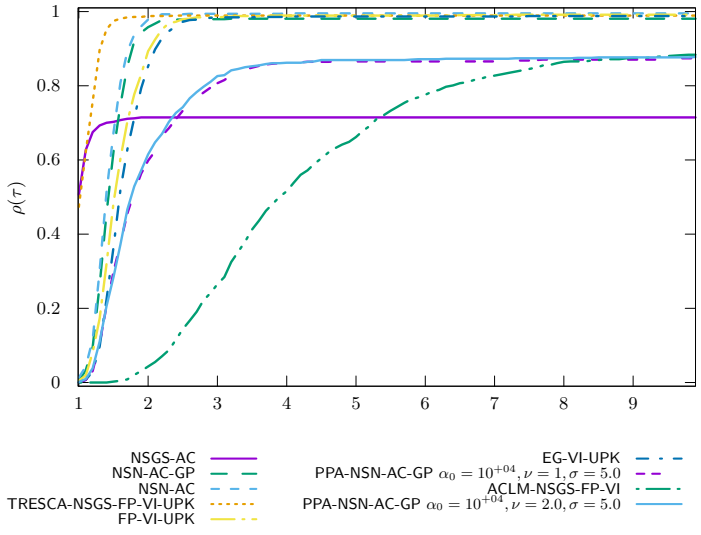


Figure 285: Chute_local_problems time COMP/zoom

Chute_local_problems precision 1.0e-08 timeout 10

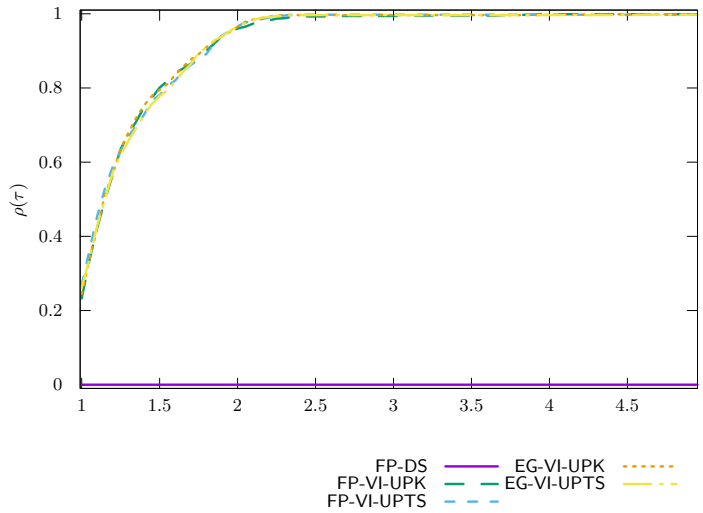


Figure 286: Chute_local_problems time VI/UpdateRule

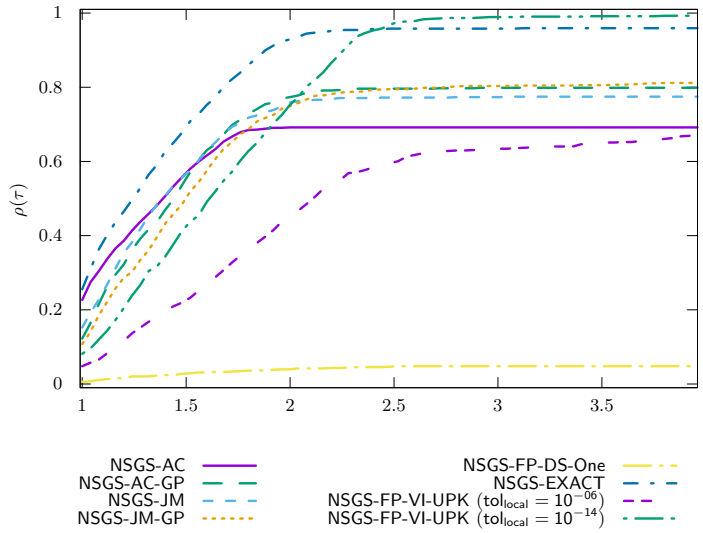


Figure 287: Chute_local_problems time NSGS/LocalSolver

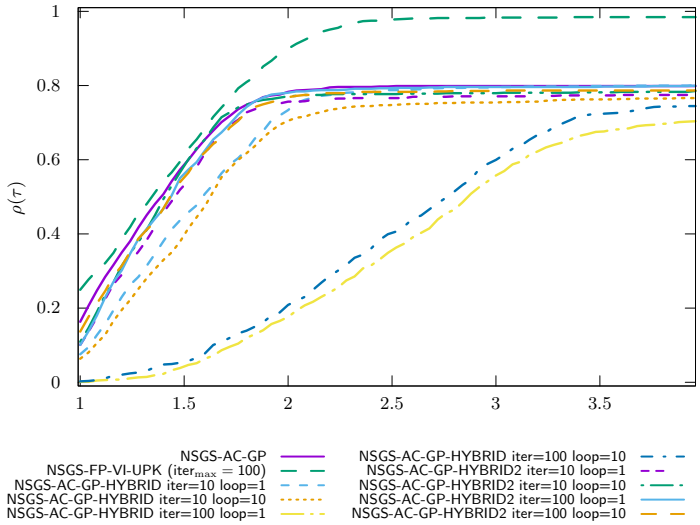


Figure 288: Chute_local_problems time NSGS/LocalSolverHybrid

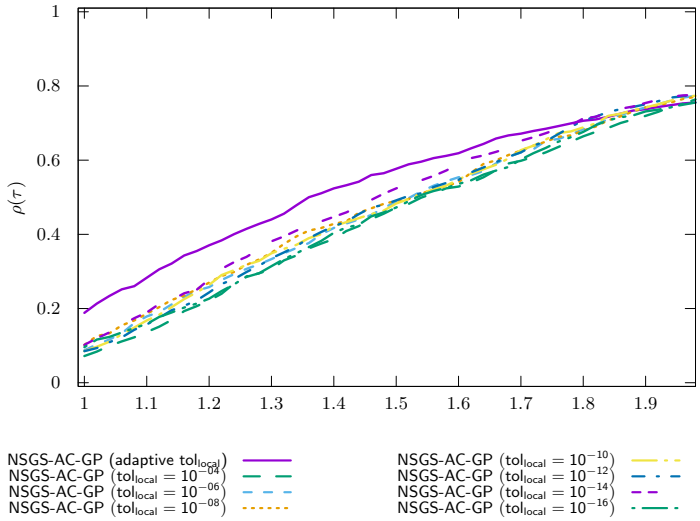


Figure 289: Chute_local_problems time NSGS/LocalTol

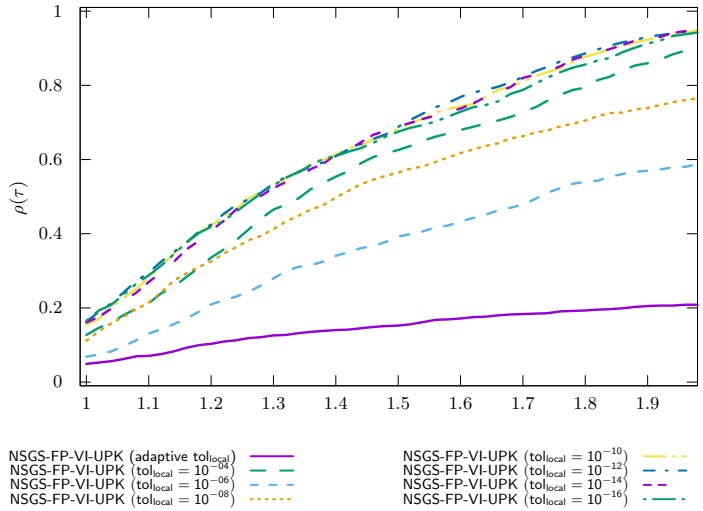


Figure 290: Chute_local_problems time NSGS/LocalTol-VI

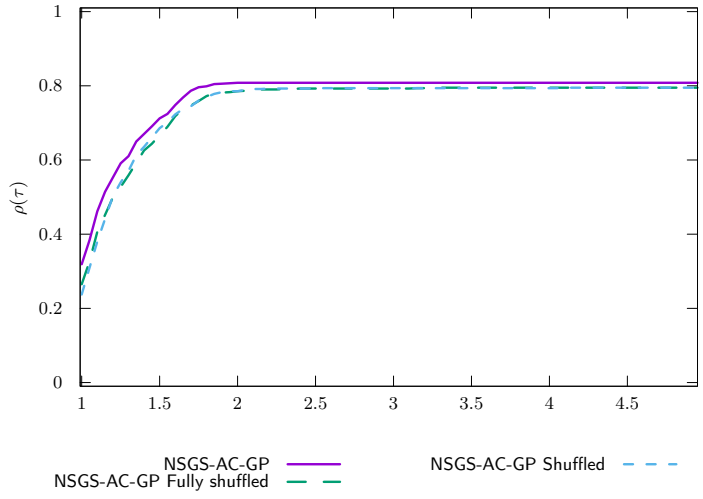


Figure 291: Chute_local_problems time NSGS/Shuffled

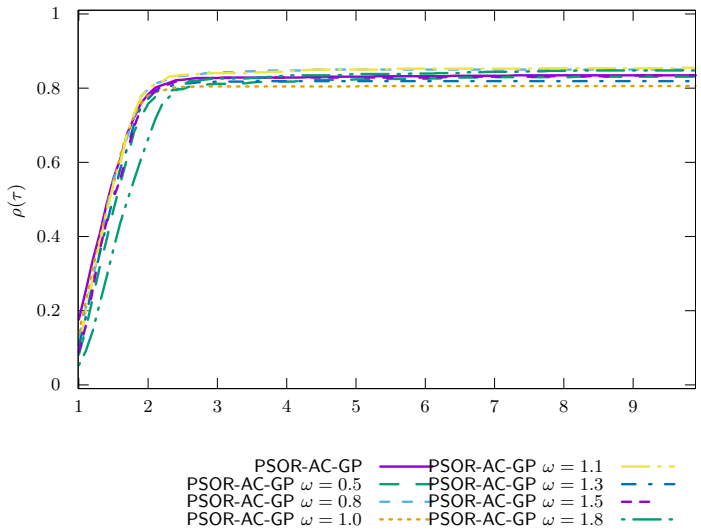


Figure 292: Chute_local_problems time PSOR

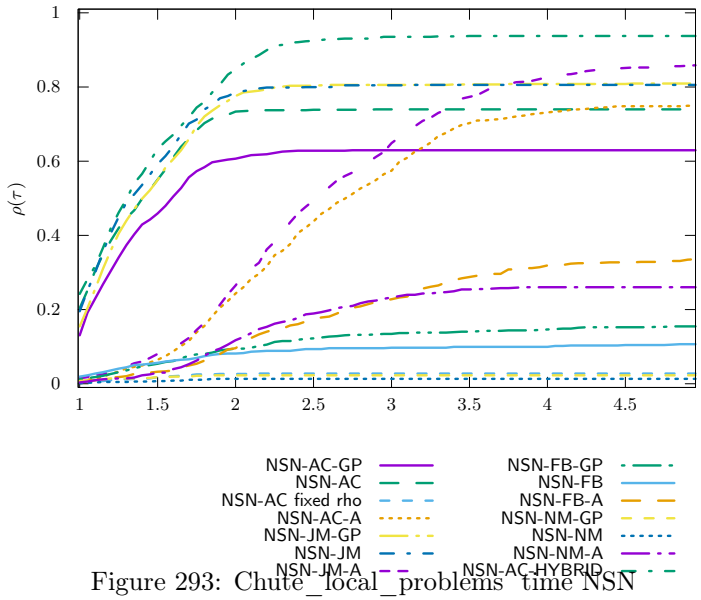


Figure 293: Chute_local_problems time NSN

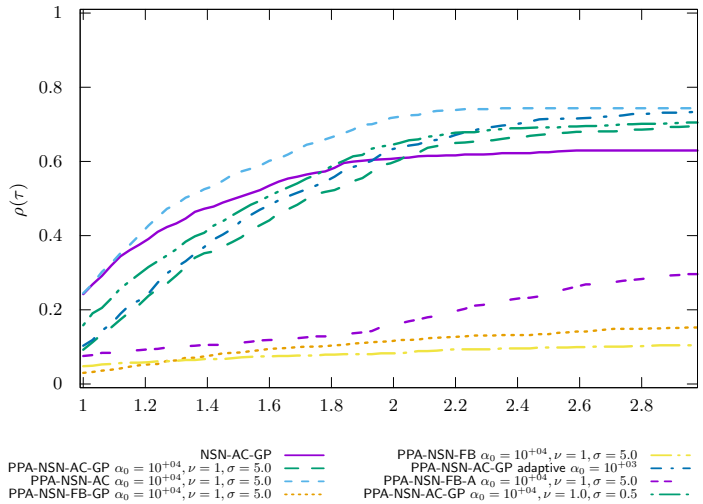


Figure 294: Chute_local_problems time PROX/NSN/InternalSolvers

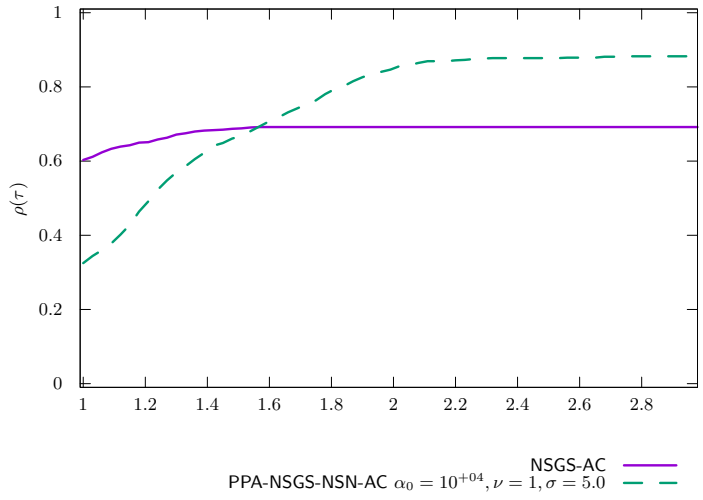


Figure 295: Chute_local_problems time PROX/NSGS/InternalSolvers

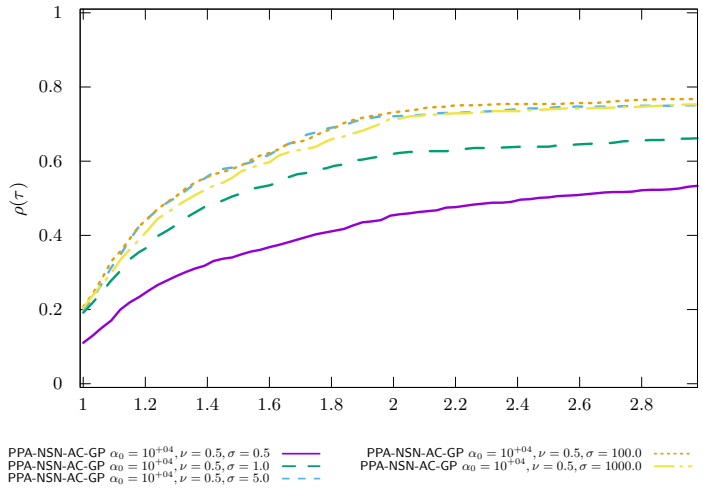


Figure 296: Chute_local_problems time PROX/Parametric studies $\nu = 0.5$

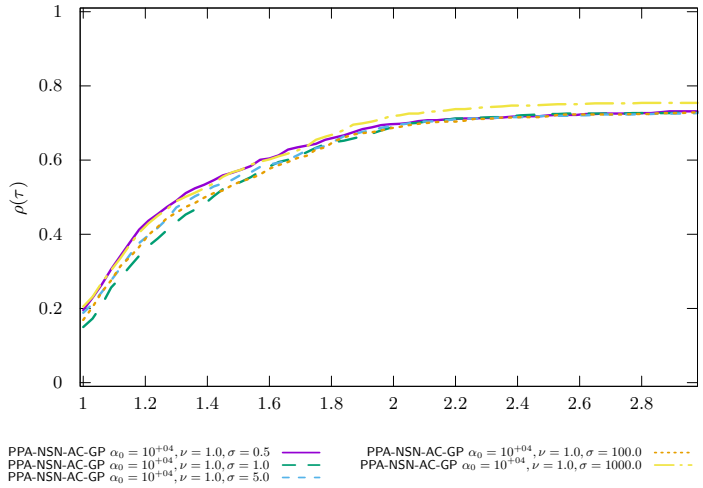


Figure 297: Chute_local_problems time PROX/Parametric studies $\nu = 1.0$

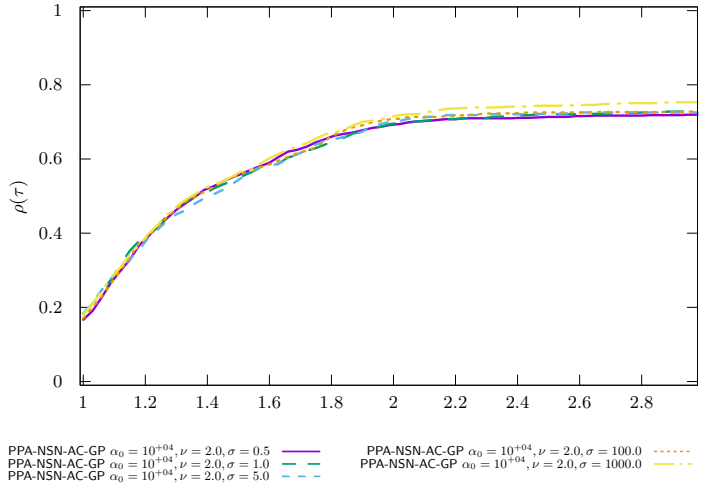


Figure 298: Chute_local_problems time PROX/Parametric studies $\nu = 2.0$

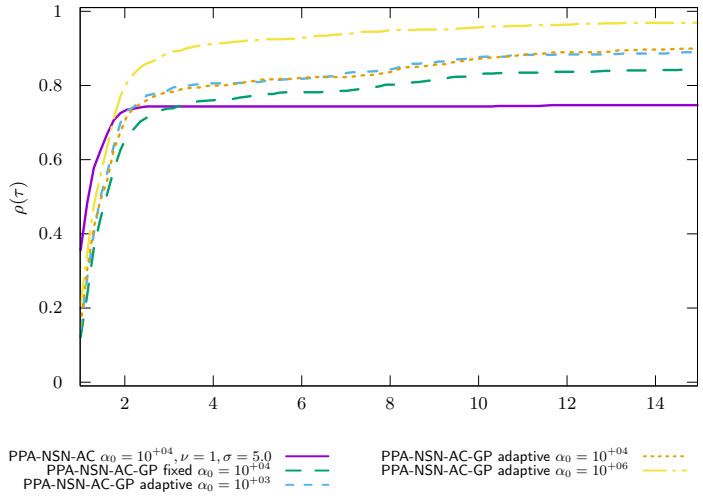


Figure 299: Chute_local_problems time PROX/Regularized problem

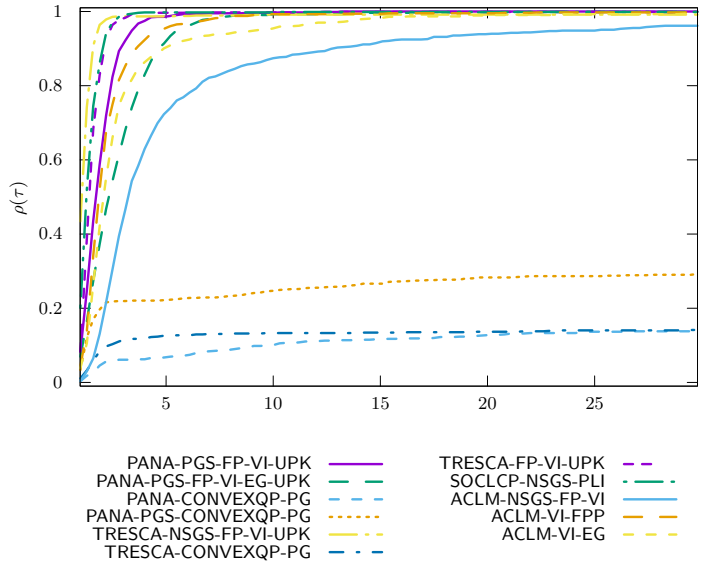


Figure 300: Chute_local_problems time OPTI

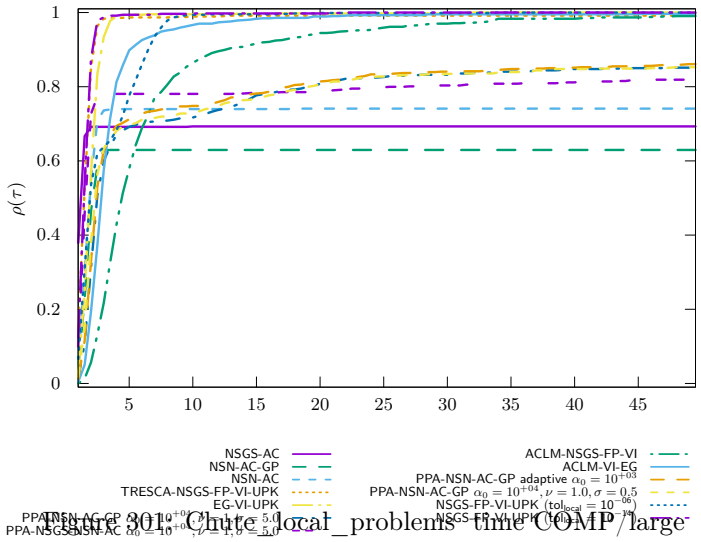


Figure 301: Chute_local_problems time COMP/large

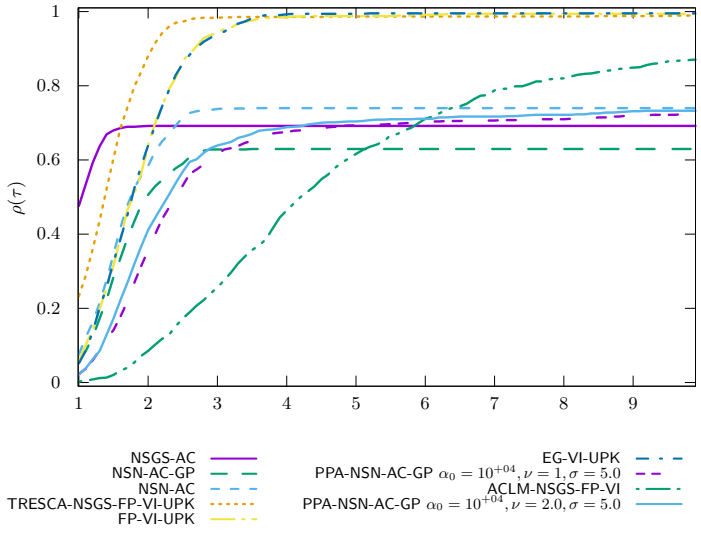
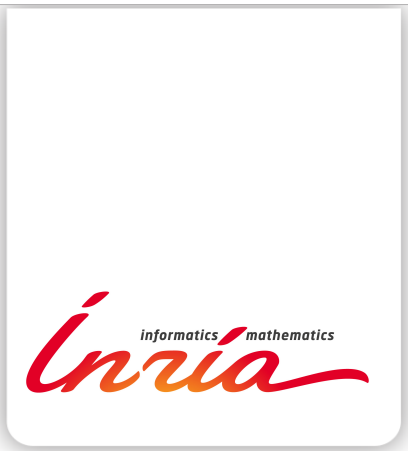


Figure 302: Chute_local_problems time COMP/zoom



RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

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

ISSN 0249-6399