# NUMERICAL METHODS AND MODELS RELEVANT TO TRANSPORTATION APPLICATIONS

THÈSE N$^{O}$ 3875 (2007)

PRÉSENTÉE LE 24 AOÛT 2007

À LA FACULTÉ DES SCIENCES DE BASE
Laboratoire transport et mobilité
SECTION DE MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Michaël THÉMANS

licencié en sciences mathématiques, Université de Namur, Belgique
et de nationalité belge

acceptée sur proposition du jury:

Prof. T. Mountford, président du jury
Prof. M. Bierlaire , directeur de thèse
Prof. T. Liebling, rapporteur
Prof. A. Sartenaer, rapporteur
Prof. N. Zufferey, rapporteur

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Lausanne, EPFL
2007

# Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Michel Bierlaire. This thesis and the related research would not have been completed without his constant help and support. His great intuition was very helpful in identifying interesting dimensions for the research and he has always been maximizing the number of good ideas to provide me during these years. I would like to thank him for all the great non-scientific moments spent at various dinners and conferences.

I am also grateful to Prof. Thomas Liebling for welcoming me in his ROSO group during four years, and for giving me the possibility of working in a pleasant atmosphere.

I would like to thank Prof. Nicolas Zufferey for our successful collaboration and for friendly discussions during lunches and breaks. I want also to thank the other members of my thesis committee, the president Prof. Thomas Mountford and Prof. Annick Sartenaer for their constructive commentaries.

This thesis was supported by the Swiss National Science Foundation grant number 205121-103598.

I wish to thank my former colleagues of the ROSO group for creating a friendly working environment and enriching my experience in Switzerland, and especially: Michela for her friendship; Rodrigue and Christophe for the numerous and memorable evenings spent together; and Frank for being a great colleague, co-author and mainly a friend.

I would also like to thank all my colleagues of the TRANSP-OR laboratory who provided a pleasant environment during the last months of this dissertation, with a special mention to Emma for our profitable collaboration as well as our pleasant discussions.

I thank all my friends from the Audipassion community and Swiss Team, with a wink to my two colleagues from Swiss Gear, Fatal Stig and Captain (not so) Slow, for nice moments of distraction and powa.

Finally I would like to thank my parents Cristine and Michel for giving me the opportunity to undertake my studies and move to Switzerland, as well as for their unbounded support. I keep all my lovely compliments for my wife My Chau. Her constant patience and support helped me a lot during these years, especially in the writing phase. I dedicate to her this dissertation with all my love.

# Abstract

In this thesis, we focus on standard classes of problems in numerical optimization: unconstrained nonlinear optimization as well as systems of nonlinear equations. More precisely, we consider two types of unconstrained nonlinear optimization problems. On the one hand, we are interested in solving problems whose second derivatives matrix is singular at a local minimum. On the other hand, we focus on the identification of a global minimum of problems which present several local minima. The increasing use of simulation tools in real applications requires solving more and more complicated problems of these classes. The main goal of this thesis is the development of efficient numerical methods, based on trust-region and filter frameworks, able to find the solution of such problems in a limited number of function evaluations. Indeed, the algorithmic developments we present have been motivated by real transportation applications in which the objective function is usually cumbersome to evaluate. The specific nonlinear optimization problems mentioned above are encountered in the estimation of discrete choice models while systems of nonlinear equations have to be solved in the context of Dynamic Traffic Management Systems (DTMS). We also dedicate a part of this dissertation to the challenging task of human behavior modeling in the context of DTMS.

First we propose a new trust-region algorithm and a new filter algorithm to solve singular unconstrained nonlinear problems. A characterization of the singularity at a local minimum is described and we present an iterative procedure which allows to identify a singularity in the objective function during the execution of the optimization algorithm. Our trust-region based algorithms make use of information on the singularity by adopting a penalty approach. Numerical results provide evidence that our approaches require less function evaluations to solve singular problems compared to classical trust-region algorithms from the literature. The CPU time to find a solution is also significantly decreased when the problem is singular.

Second we present a new heuristic designed for nonlinear global optimization, based on the variable neighborhood search from discrete optimization within which we use a trust-region algorithm from nonlinear optimization as local search procedure. The algorithm we propose is able to prematurely stop the local search as soon as it does not look promising. The neighborhoods and the neighbors selection are based on information about the curvature of the objective function. Intensive numerical tests illustrate that our method is able to significantly reduce the average number of function evaluations compared to existing heuristics in the literature of nonlinear global optimization. Important improvements are also obtained in terms of success rate as well as CPU time.

Third we design a new secant method for systems of nonlinear equations. The proposed algorithm uses a population of previous iterates and the linear model of the system is calibrated using a least squares approach. We also propose two globalization techniques for quasi-Newton methods in this context, namely a linesearch framework and a linesearch-filter approach. Our algorithm exhibits a faster convergence as well as a better robustness compared to secant methods from the literature. Globalization strategies are shown to highly increase the robustness of considered secant methods. Moreover, the combination of our algorithm with these strategies gives rise to an algorithmic method which is competitive with Newton-Krylov methods both in terms of robustness and efficiency.

Fourth we present a real application of discrete choice models in the context of DTMS. The models are designed to capture the response of Swiss drivers to real-time traffic information. We are interested in drivers' decisions in terms of both route and mode choices when traffic information is available before the trip starts while we focus on route choice when traffic information is available during the trip. The "en-route" model is a mixture of binary logit model with panel data while "pre-trip" models are nested logit models. These models are estimated with the BIOGEME software developed by Bierlaire (2003). Estimation results are deeply analyzed and discussed, and models are implemented in a simulator which predicts drivers' behavior in specific scenarii.

We conclude this thesis by a review of the main results and we make some comments about promising tracks for future research.

Keywords: numerical optimization, trust-region algorithms, filter, secant methods, discrete choice models, transportation applications.

# Résumé

Dans cette thèse, nous nous concentrons sur des classes standards de problèmes en optimisation numérique: l'optimisation non-linéaire sans contrainte ainsi que les systèmes d'équations non-linéaires. Plus précisément, nous nous intéressons à deux types de problèmes d'optimisation. D'une part, nous souhaitons résoudre des problèmes dont la matrice des dérivées secondes est singulière en un minimum local. D'autre part, nous nous concentrons sur l'identification d'un minimum global de problèmes qui présentent plusieurs minima locaux. De part l'utilisation sans cesse croissante d'outils de simulation dans les applications réelles, ces problèmes sont de plus en plus compliqués à résoudre en pratique. Le but principal de cette thèse est de développer des algorithmes de région de confiance et de filtre efficaces et capables de trouver la solution de tels problèmes en un nombre limité d'évaluations de fonction. En effet, les développements algorithmiques que nous présentons ont été motivés par des applications réelles en transport dans lesquelles la fonction-objectif est généralement coûteuse à évaluer. Les problèmes d'optimisation non-linéaire spécifiques mentionnés ci-dessus sont rencontrés dans l'estimation de modèles de choix discret tandis que des systèmes d'équations non-linéaires doivent être résolus dans le contexte de systèmes dynamiques de gestion du trafic (DTMS). Nous consacrons également une partie de cette thèse au défi que constitue la modélisation du comportement humain dans le contexte de DTMS.

Premièrement, nous proposons un nouvel algorithme de région de confiance et un nouvel algorithme de filtre pour résoudre des problèmes d'optimisation non-linéaire sans contrainte singuliers. Nous donnons une caractérisation de la singularité en un minimum local et nous présentons une procédure itérative qui permet d'identifier une singularité dans la fonction-objectif pendant l'exécution de l'algorithme d'optimisation. Nos algorithmes utilisent cette information sur la singularité en adoptant une approche par pénalité. Les résultats numériques prouvent que nos algorithmes nécessitent moins d'évaluations de fonction pour résoudre des problèmes singuliers, en comparaison des algorithmes classiques de région de confiance de la littérature. Le temps de calcul nécessaire pour trouver une solution est aussi sensiblement réduit lorsque le problème est singulier.

Deuxièmement, nous présentons une nouvelle heuristique conçue pour l'optimisation non-linéaire globale, basée sur une recherche à voisinage variable issue de l'optimisation discrète dans laquelle nous utilisons un algorithme de région de confiance provenant de l'optimisation non-linéaire comme procédure de recherche locale. L'algorithme que nous proposons est capable d'interrompre prématurément la recherche locale dès qu'elle ne semble pas prometteuse. Les voisinages et la sélection des voisins sont basés sur l'information à propos de la courbure de la fonction-objectif. Des tests numériques intensifs montrent que notre méthode permet de réduire significativement le nombre moyen d'évaluations de fonction en comparaison d'heuristiques existantes dans la littérature d'optimisation non-linéaire globale. Des améliorations conséquentes sont également obtenues aussi bien en terme de taux de succès qu'en terme de temps de calcul.

Troisièmement, nous concevons une nouvelle méthode sécante pour les systèmes d'équations non-linéaires. L'algorithme proposé utilise une population d'itérés précédents et le modèle linéaire du système est calibré par moindres carrés. Nous proposons également deux techniques de globalisation pour les méthodes quasi-Newton dans ce contexte, à savoir une technique de recherche linéaire et une technique de filtre combinée à une recherche linéaire. Notre algorithme montre une convergence plus rapide ainsi qu'une robustesse accrue en comparaison des méthodes sécantes de la littérature. Les stratégies de globalisation permettent une augmentation flagrante de la robustesse des méthodes sécantes considérées. De plus, la combinaison de notre algorithme avec ces stratégies donne lieu à une méthode algorithmique qui est compétitive avec les méthodes de type Newton-Krylov en termes de robustesse et d'efficacité.

Finalement, nous présentons une application réelle de modèles de choix discret dans le contexte de DTMS. Les modèles sont conçus pour appréhender la réponse des conducteurs suisses face à de l'information routière en temps réel. Nous nous intéressons aux décisions des conducteurs en termes de choix de route et de mode de transport lorsque de l'information routière est disponible avant que le trajet ne commence alors que nous nous concentrons sur le choix de route lorsque cette information est disponible au cours du trajet. Le modèle "en route" est un modèle logit binaire mixte avec des données de type "panel" tandis que les modèles "pre-trip" sont des modèles logit emboîtés. Ces modèles sont estimés avec le logiciel BIOGEME développé par Bierlaire (2003). Les résultats d'estimation sont analysés et discutés en détail, et les modèles sont implémentés dans un simulateur qui permet de prédire le comportement de conducteurs dans des scénarii spécifiques.

Nous terminons cette dissertation par un récapitulatif des principaux résultats obtenus et nous commentons des pistes prometteuses pour la recherche future.

Mots-clés: optimisation numérique, algorithmes de région de confiance, filtre, méthodes sécantes, modèles de choix discret, applications en transport.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## Contents

## 1.1 General framework of the research

More and more complex mathematical models have been proposed during the last decades in a wide range of domains, such as transportation, engineering, economics, finance, industry, etc. In addition to an increased complexity, the developments in the power of computers have also motivated the use of advanced simulation tools in order to represent the reality as close as possible while still simplifying it. In consequence, standard classes of problems arising in numerical optimization may become more and more problematic to solve. This is mainly due to the complicated form of the involved functions. Moreover, these functions may be very expensive to evaluate in terms of computational costs, as they require simulation tools in most cases.

This thesis mainly focuses on two fundamental problems of numerical optimization, namely:

- unconstrained nonlinear optimization
- systems of nonlinear equations

On the one hand, we are interested in specific optimization problems in Chapters 2 and 3, that is:

- problems whose second derivatives matrix is singular at the local minimum
- problems which present several (and often many) local minima and the challenge is to identify a global one.

The development of specific algorithms dedicated to solve such optimization problems has been motivated by the maximum likelihood estimation of discrete choice models widely used in transportation.

On the other hand, we consider general systems of nonlinear equations arising in the context of Dynamic Traffic Management Systems (DTMS) and we propose a new algorithm for solving such problems in Chapter 4. Moreover, in Chapter 5, we propose behavioral models for real-time transportation applications such as DTMS.

To conclude the presentation of the thesis framework, we emphasize on the fact that the common goal of all algorithmic methods proposed in this dissertation is to reduce the number of function evaluations necessary to get a solution of the considered problems.

## 1.2 Origin of the thesis

### 1.2.1 Maximum likelihood estimation (MLE) of discrete choice models (DCM)

Discrete choice models are mathematical models used to analyze and predict the behavior of individuals when faced to choice situations (such as decisions about transportation mode, route choice, etc). These econometric models play an important role in transportation analysis. The theoretical foundations of discrete choice models (and more specifically, random utility models) had already been defined in the seventies (Ben-Akiva, 1973, Williams, 1977, McFadden, 1978) with the multinomial logit model, the multinomial probit model, the nested logit model, and the generalized extreme value model. However, only the multinomial logit model and the nested logit model have been intensively used by practitioners during almost three decades. These models are relatively easy to estimate, as their associated log-likelihood function has nice properties (globally concave for the multinomial logit model, concave in a subspace of parameters for the nested logit model). Therefore, the use of the classical Newton-Raphson optimization algorithm is most of the time appropriate.

Recent advances in discrete choice models are following two complementary tracks. Firstly, more "logit-like" models within the generalized extreme value family have been proposed and used (see, for instance, Vovsha, 1997, Wen and Koppelman, 2001, Bierlaire, 2002, Papola, 2004, Bierlaire, 2006b and Daly and Bierlaire, 2006). Secondly, the increasing power of computers has motivated the use of mixtures of logit models, where the normal distribution of some parameters requires simulation methods to compute the probability model (McFadden and Train, 2000, Bhat, 2001, Train, 2003). Actually, mixtures of GEV models start to be proposed as well in the literature (see Bhat and Guo, 2004, Hess et al., 2004, Hess et al., 2005a and Hess et al., 2005c). Finally, discrete mixtures of GEV models are also investigated (see Hess et al., to appear).

Estimating those models, that is computing the maximum log-likelihood, becomes more and more problematic. Firstly, the objective function becomes highly nonlinear and non-concave. Secondly, the computational cost of evaluating the objective function and its derivatives becomes significantly high. Thirdly, the optimization problem to be solved might be singular due to model overspecification or mis-specification by the modeler, causing the optimization algorithm to converge slowly and highly increasing the estimation time. Fourthly, the complexity of the model often requires constraints on the parameters, in order to obtain meaningful values, or to overcome model overspecification.

Classical unconstrained optimization algorithms can no longer be applied and we thus need optimization algorithms able to deal with singularities in the objective function (that is the log-likelihood function to be maximized) on the one hand and with several, and possibly many, local optima on the other hand. In Chapter 2, we design trust-region based optimization algorithms able to efficiently solve problems which are singular at a local minimum. Chapter 3 presents a new algorithmic method for nonlinear global optimization which is capable of quickly identifying a global minimum of an unconstrained nonlinear optimization problem whose objective function presents several local minima.

In the following, we discuss in details the motivations which gave rise to the algorithmic developments presented in Chapters 2 and 3.

### Singularity issues

In the context of discrete choice models estimation, and more generally econometric models, the causes of the singularity can actually be multiple, for instance:

- The theoretical model contains too many parameters and not all of them are identifiable (in this case we speak about model overspecification).
- There may be a mis-specification in the utilities due to the modeler.
- The specification of the utility functions contains more parameters than the data allows to estimate, due to a lack of variability.

In the first case, the singularity is structural in the sense that it is due to the theoretical model used. In the last two cases, the source of the singularity comes from a poor model specification by the modeler, which frequently happens during the model development phase, or it is due to data limitations. In other terms, the singularity can be structural when the identification issue is related to the parameters of the error terms or it can be contextual when the identification issue comes from parameters of the deterministic part of the utility functions.

When the singularity is structural, the log-likelihood function can be shown to be singular at each point, that is the second derivatives matrix is singular at each iterate, while it is only singular at a local optimum when the singularity is contextual.

Some of the algorithms available in the optimization package BIOGEME dedicated to discrete choice models estimation and developed by Bierlaire (2003) are already robust to face with structural singularities. Indeed, one can show that a trust-region based algorithm can deal with singularity at each point when the trust-region subproblem is solved by using a truncated conjugate gradient method.

This is the reason why we focus in this dissertation on singularities in the objective function only at the local optimum of the optimization problem. In this case, even if the second derivatives is singular only at the minimum, the convergence of the overall sequence of iterates is significantly deteriorated for all Newtonian methods. For instance, while Newton's method is known to exhibit a quadratic rate of local convergence to a local minimizer when the second derivatives matrix is non-singular at this local minimizer, Griewank and Osborne (1983) have shown that the iterates produced are at best linearly convergent when this matrix is singular (even if the second derivatives matrix is non-singular at all iterates). In addition to the fact that the convergence of the overall sequence of iterates is significantly deteriorated, evaluating the log-likelihood function can be expensive in terms of computational cost and, consequently, the estimation time can be huge. Furthermore, when solving singular problems, standard methods can encounter numerical problems. Finally, the variance-covariance matrix of the estimates cannot be obtained from the inversion of the second order derivatives matrix of the log-likelihood function. As a consequence, statistical tests of these estimates are no more available, meaning that it is not possible to assess the quality of the calibrated model.

We thus propose in Chapter 2 an optimization algorithm which is able to significantly reduce the number of function evaluations necessary to solve problems which are singular at the local minimum. Important gain in CPU time can be expected when estimating advanced discrete choice models involving such singularity issues.

### Non-concavity issues

The major concern in Chapter 3 is about the non-concavity of the log-likelihood function associated with advanced discrete choice models. Indeed, the advanced models discussed above are usually associated with complicated form of the log-likelihood function presenting several (and often many) local optima. In this case, we thus need specific optimization algorithms designed to deal with many local optima and able to identify a global one.

There exist several powerful softwares for discrete choice models estimation like, for instance, BIOGEME (see Bierlaire, 2003), ALOGIT and LIMDEP. However, none of them is able to compute the global optimum of non-concave problems but only a local optimum. One of the main interests of the work presented in Chapter 3 is to propose a new algorithm which could overcome this drawback.

Note that finding the global optimum of a nonlinear optimization problem is also of major importance in many other transportation applications such as traffic equilibrium problems. These problems aim to assign flows on a transportation network such that a given objective is optimized (for instance, the total travel time on the network in the context of a system optimum equilibrium). In such contexts, only the global optimum of the problem will provide meaningful optimal values for the flows on the network. Indeed, a local optimum will in general not be interesting from the application viewpoint.

We thus propose in Chapter 3 an efficient optimization algorithm able to identify a global minimum of an unconstrained nonlinear optimization problem.

### 1.2.2 Dynamic Traffic Management Systems (DTMS)

As we saw in the previous subsection, estimating advanced discrete choice models requires the development of specific optimization algorithms which can take into account the specificities of the corresponding optimization problems. Discrete choice models are powerful tools to predict demand in various complex transportation contexts as they allow to obtain a disaggregate representation of the demand in these specific contexts, where individuals are considered with their socio-economic characteristics as well as their decisions in terms of mode choice, route choice, etc.

However, efficient demand models are not sufficient to develop efficient Dynamic Traffic Management Systems (DTMS) able to considerably improve general traffic conditions. It is important to take into account the economic interpretation of transportation, that is the interaction between demand and supply. This fundamental concept is critical for understanding, designing, and most importantly, managing Intelligent Transportation Systems (ITS). Actually, the standard framework for DTMS combines three elements: advanced surveillance systems collecting real-time traffic data, Advanced Traffic Management Systems (ATMS) and Advanced Traveler Information Systems (ATIS). On the one hand, the ATMS affects the network supply, imposing restrictions and constraints on traffic flows, predicting traffic congestion, and providing alternative routing instructions to vehicles to improve the network performance. On the other hand, the ATIS affects the transportation demand by providing historical, real-time and predictive information to support drivers travel decisions before and during their trip. ATIS support includes Radio Data System (RDS), GPS navigation systems, and variable message signs (VMS).

An efficient application of these systems (ATMS/ATIS) using DTMS must be based on implicit or explicit simulation of the interaction between demand (vehicles) and supply (network). This interaction can be formulated as a system of nonlinear equations. More generally, many transportation problems can be formulated as systems of nonlinear equations. One can cite traffic equilibrium problems, consistent anticipatory route guidance (see Bottom, 2000 and Crittin, 2004) and many others.

As these problems involve most of the time simulators in transportation applications, the systems to be solved are usually expensive to evaluate. It is mandatory to solve them using efficient algorithms able to reduce the number of function evaluations and the CPU time to get a solution. These applications have motivated the algorithmic developments presented in Chapter 4. We present a new multi-iterate secant method for solving systems of nonlinear equations which could be integrated in DTMS.

Chapter 5 presents discrete choice models in the context of DTMS able to explicitly capture and predict demand in transportation systems in which traffic information is available in real-time. These models account for the response of drivers to advanced information about traffic conditions provided by telematic technologies, such as the ones used in Intelligent Transportation Systems (ITS). As the impact of ITS on drivers' behavior in terms of route and mode choice can be correctly apprehended, the presented models could consequently be integrated in Dynamic Traffic Management Systems (DTMS) which combine advanced information and control technologies provided by ITS with decision-aid tools.

## 1.3 Thesis contributions

The main contributions of this thesis lie in the domain of numerical optimization. Two of the main classes of problems are considered in the dissertation, namely unconstrained nonlinear optimization problems as well as systems of nonlinear equations.

We identify four important contributions of this thesis:

- New trust-region based algorithms able to deal with singularities in unconstrained nonlinear optimization. The proposed methods are shown to significantly outperform classical trust-region and filter algorithms in terms of efficiency, computational time and robustness when the considered problems are singular at a local minimum. They are able to identify a singularity during the course of the algorithm and to use this information in order to accelerate the convergence speed. Handling singularities in the objective function is difficult because the efficiency and convergence theory of existing methods is characterized by the curvature of the objective function, which is null for singular problems.

- A new heuristic for nonlinear global optimization. The new algorithm is intensively tested showing evidence of its superiority on existing heuristics as the probability of finding the global optimum of a general unconstrained nonlinear problem is increased while reducing the average number of function evaluations necessary to achieve this objective. Important improvements in computational time are also obtained when considering problems with many variables and/or a cumbersome objective function.

- A multi-iterate secant method to solve systems of nonlinear equations. Numerical experiments illustrate better capababilities in robustness and efficiency in comparison to classical quasi-Newton methods of the literature. Two globalization techniques are derived and shown to highly increase the robustness of quasi-Newton methods for solving difficult problems. The combination of the proposed secant method with these techniques gives rise to an algorithm which is competitive in terms of robustness and efficiency compared to Newton-Krylov methods which have been proved to be efficient derivatives-based methods to solve systems of nonlinear equations.

- The development of behavioral models in Switzerland in the context of real-time applications. This constitutes the first step in Switzerland toward the development of DTMS. Contrarily to behavioral models proposed in the literature to capture the response of drivers to traffic information, we present two originalities. Firstly, we extend the analysis to both radio information and information coming from VMS, and we use different samples. Secondly, we do not only focus on en-route switching-decisions but we also consider pre-trip mode choice decisions. In addition, the simulator integrates both en-route and pre-trip models allowing to assess the impact of traffic information in numerous scenarii.

## 1.4 Thesis organization

Chapter 2 proposes new trust-region based algorithms for unconstrained nonlinear optimization designed to deal with singularities in the objective function. More precisely, a trust-region algorithm as well as a filter method are presented to solve problems whose second derivatives matrix is singular at a local minimum, violating one of the main assumptions in convergence analysis of most newtonian methods for unconstrained optimization. In this chapter, we give a theoretical characterization of such a singularity and we propose an iterative procedure which allows to identify a singularity in the objective function during the course of the optimization algorithm. Consequently, we derive two new optimization algorithms able to make use of this information on the singularity by adopting a penalty approach. Numerical tests highlight the significant improvement of the proposed algorithms compared to classical trust-region and filter methods in the literature. Our new methods are able to significantly reduce the number of function evaluations and the CPU time necessary to converge to a solution when the problem is singular.

Chapter 2 contains mainly the developments proposed in Bierlaire and Thémans (2006a). This paper has been submitted to the *European Journal of Operational Research.*

Chapter 3 is dedicated to a new heuristic designed for nonlinear global optimization. This algorithm consists in a Variable Neighborhood Search (VNS) heuristic within which a trust-region based algorithm is used as local search. It allows for a better use of information on the objective function compared to existing methods as we approximate derivatives of the objective function up to second order. The algorithm is also able to prematurely interrupt the local search if the area of search does not look promising, due to proximity of already identified local minima or to insufficient improvement in the objective function. The advantage is to limit the number of iterations of the local search and the overall number of function evaluations. Finally, the neighborhoods and the neighbors selection procedure are based on the geometry of the objective function and the use of curvature information allows the iterates to escape from local minima. Numerical experiments clearly demonstrate the superiority of our approach compared to existing methods of the literature. Intensive tests show that the proposed VNS algorithm outperforms its competitors both in terms of efficiency and success rate. It significantly reduces the average number of function evaluations necessary to get a global minimum of a general unconstrained nonlinear optimization problem. On large-scale problems, the gain obtained in CPU time is also very important.

The main parts of Chapter 3 are inspired from Bierlaire et al. (2007), which has been submitted to *INFORMS Journal on Computing.*

Chapter 4 describes a new secant method for solving systems of nonlinear equations. The proposed algorithm uses a population of previous iterates, generalizing state-of-the-art quasi-Newton methods. We also prefer a least squares approach to calibrate the linear model of the system rather than exact interpolation. Our method is shown to lead to an update formula. Finally, we develop two globalization techniques, using a linesearch framework for the first one and a linesearch-filter framework for the second one. Numerous numerical tests are performed against classical secant methods of the literature, using both undamped and damped versions of the challenged methods. Our new algorithm, called GSM, exhibits a faster convergence as well as a better robustness compared to its competitors. Proposed globalization strategies are shown to highly increase the robustness of considered secant methods. When GSM is combined with our globalization techniques, it is even shown to behave in a similar way to Newton-Krylov methods.

Chapter 4 is mainly constituted from ideas presented in Bierlaire et al. (to appear) which has been accepted for publication in *European Journal of Operational Research.* The ideas of the GSM algorithm has been originally proposed by Bierlaire and Crittin (2003).

While Chapters 2-4 are related to algorithms, Chapter 5 deals with modeling as it involves the development of discrete choice models for real-time transportation applications. Indeed, we present behavioral models designed to capture the response of Swiss drivers to real-time traffic information. We are interested in both route and mode choices when traffic information is available before the trip starts while we focus on route choice when traffic information is available during the trip. First we present the en-route model which consists in a mixture of Binary Logit model with panel data to analyze drivers's decisions when information about traffic conditions is provided by the means of Radio Data System (RDS) or Variable Message Signs (VMS). Second we propose Nested Logit models in order to capture their behavior when they are aware of traffic conditions before starting their trip. All models are estimated with the BIOGEME software developed by Bierlaire (2003) and estimation results are deeply analyzed and discussed. Finally, models are implemented in a simulator which predicts drivers' behavior in specific scenarii and allows for sensitivity analysis of the demand with regard to variation of different parameters.

The methodology presented in Chapter 5 consists for most parts of results presented in Bierlaire et al. (2006), which has been published in *European Transport*. It is also inspired from a report for the Swiss Federal Office of Roads (see Bierlaire and Thémans, 2006b).

The conclusions of the work presented in this dissertation are summarized in Chapter 6. We provide a review of the main results of the research performed during this thesis and we discuss future developments related to this research.

We conclude this introduction by a summary of the work presented in the next chapters of the dissertation. For each chapter, Table 1.1 describes the type of problem for which numerical methods/models we propose are designed as well as the application of interest (discussed in Section 1.2).

| Chapter | Problem | Application |
|---|---|---|
| 2 | Unconstrained nonlinear optimization | MLE of DCM |
| 3 | Nonlinear global optimization | MLE of DCM |
| 4 | Systems of nonlinear equations | Equilibrium problems in transportation |
| | | DTMS |
| 5 | Behavior modeling | DTMS |

Table 1.1: Outline of the chapters

# Chapter 2

# Singularities in nonlinear optimization

## Contents

## 2.1 Introduction

We propose new trust-region based optimization algorithms for solving unconstrained nonlinear problems whose second derivatives matrix is singular at a local minimum. We give a theoretical characterization of the singularity in this context and we propose an iterative procedure which allows to identify a singularity in the objective function during the course of the optimization algorithm, and artificially adds curvature to the objective function. Numerical tests are performed on a set of unconstrained nonlinear problems, both singular and non-singular. Results illustrate the significant performance improvement compared to classical trust-region and filter algorithms proposed in the literature.

This chapter contains mainly the developments proposed in Bierlaire and Thémans (2006a). This paper has been submitted to the *European Journal of Operational Research.*

## 2.2 Motivation

We consider a nonlinear unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \tag{2.1}$$

where f is a twice differentiable function, possibly nonconvex. The most efficient methods to identify a local minimum of (2.1) are variants of Newton's method, based on globalization techniques such as linesearch and trust-region methods, as described in many textbooks including Dennis and Schnabel (1996), Nocedal and Wright (1999), Bertsekas (1999) and Bierlaire (2006a).

The convergence analysis of these algorithms assumes that the curvature of the objective function at the solution $x^*$ is bounded away from 0, that is the second derivatives matrix is positive definite at $x^*$.

However, this major assumption cannot always be guaranteed in practice. This is typically the case for the maximum likelihood estimation of the parameters of econometric models. In this context, the source of singularity is twofold. On the one hand, a lack of variability in the data may preclude the identification of some parameters. On the other hand, some advanced models require complicated normalization which cannot be performed, imposing the estimation of an unidentified model (see Walker, 2001 and Thémans and Bierlaire, 2006).

In the presence of singularity, not only the convergence theory cannot be applied as such anymore, but significant deterioration of the algorithm performance is observed. In this chapter, we propose a variant to existing trust-region and filter-trust-region methods in order to deal with this issue.

## 2.3 Literature review

The convergence theory of Newton-like methods guarantees local quadratic convergence if the eigenvalues of the second derivatives matrix of the iterates are bounded away from 0. Griewank and Osborne (1983) have shown that if a problem is singular for an algorithm, the iterates produced are at best linearly convergent (even if the second derivatives matrix is singular only at the solution, and not at all iterates). Furthermore, when solving singular problems, standard methods can encounter numerical problems as the curvature of the function converges toward zero.

In the literature, singular problems have been mainly considered in the context of solving systems of nonlinear equations (see, for instance, Decker and Kelley, 1980, Decker et al., 1983, Griewank and Osborne, 1983, Schnabel and Frank, 1984, Griewank, 1985 and Izmailov and Solodov, 2002). Decker and Kelley (1980) have worked on the theoretical implications of singularity in the Jacobian of the system at a local solution. They have shown that the convergence deteriorates and can be proved to be asymptotically linear of ratio $\frac{2}{3}$ for some classes of singular systems. Griewank and Osborne (1983) have analyzed the behavior of Newton's method near singularities in the Jacobian. In the singular case, Newton's method can either converge with a limiting linear ratio, or diverge from arbitrarily close starting points or even behave chaotically. Decker et al. (1983) have analyzed in details the linear convergence rates of Newton's method on several classes of singular problems. They also propose a modification of the method, constraining iterates in regions where the Jacobian is invertible, which allows to restore the quadratic rate of convergence for some of these classes of singular problems. Schnabel and Frank (1984) have introduced a new class of methods, called tensor methods, for solving systems of nonlinear equations. Tensor methods are particularly well adapted when the Jacobian matrix at the solution is singular or badly conditioned. The main idea is to consider a quadratic model instead of using the classical linear model as in Newton-like methods. The second-order term of this new model is determined such that the model interpolates the function values at several previous iterates, as well as the function value and its gradient value at the current iterate. Griewank (1985) also proposed a quadratical model in order to deal with singular solutions. Moreover, two modifications of the Newton's recurrence scheme are proposed to solve singular problems more efficiently. Izmailov and Solodov (2002) have proposed a new algorithm to solve singular problems such as smooth reformulations of nonlinear complementarity problems. The idea is to regularize a singular solution $\bar{x}$ by adding another term to the left-hand size, which vanishes at $\bar{x}$ (so that $\bar{x}$ remains a solution), and such that its Jacobian at $\bar{x}$ "compensates" for the singularity. They suggest to base this extra term on the information about the derivative of the system.

In the context of unconstrained optimization, Schnabel and Chow (1991) have proposed to use tensor methods as an adaptation of tensor methods for systems of nonlinear equations. Tensor methods dedicated to optimization construct a fourth-order model using third and fourth derivatives tensors of the objective function $f$. These higher-order derivatives allow to deal with singularity in the second derivatives matrix at local minima.

In the next section, we give a characterization of the singularity and a procedure which allows to identify this singularity during the course of the optimization algorithm. We present in Section 2.5 a class of algorithms designed to deal with singular problems in an efficient way. Based on the trust-region framework, it is able to accomodate advanced variants based on preconditioning and filter.

## 2.4 Characterization and identification of the singularity

Due to the possible non-convexity of $f$ in (2.1), the objective function may exhibit several local optima. Some of them may correspond to a singular second derivatives matrix, but not necessarily all of them. We are interested here in the case where a given algorithm $a$ converges to a singular local minimum. Consequently, we say that problem (2.1) is *singular* for algorithm $a$ if the algorithm generates a sequence $x_k$, converging to $x^*$ such that $\nabla f(x^*) = 0$, $\nabla^2 f(x^*)$ is semi positive definite, and $\nabla^2 f(x^*)$ is singular.

We denote by $A$ the $n \times m$ matrix characterizing the singularity. Its range is the eigensubspace associated with the null eigenvalues of the second derivatives matrix $\nabla^2 f(x^*)$. Formally, let's assume that $\nabla^2 f(x^*)$ has $m < n$ null eigenvalues, so that its Schur decomposition is

$$\nabla^2 f(x^*) = (A\ B) \begin{pmatrix} 0 & 0 \\ 0 & \Lambda_2 \end{pmatrix} \begin{pmatrix} A^T \\ B^T \end{pmatrix} = B\Lambda_2 B^T$$

where $A \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{n x (n-m)}$ are orthonormal, and the columns of $A$ are the eigenvectors corresponding to 0 eigenvalues. In this case, any direction in the range of $A$ does not modify (at least asymptotically) the value of $f$. Indeed, for an arbitrary $s \in \mathbb{R}^m$,

$$\begin{aligned} f(x^* + As) &= f(x^*) + \nabla f(x^*)^T As + \tfrac{1}{2} s^T A^T B \Lambda_2 B^T As + o(\|s\|^2) \\ &= f(x^*) + o(\|s\|^2), \end{aligned}$$

as $\nabla f(x^*) = 0$ and $A \perp B$. Invoking the fundamental theorem of linear algebra, we know that the subspace orthogonal to $\mathrm{Im}(A)$ is the null-space of $A^T$, that is $\mathrm{Im}(A)^\perp = \ker(A^T)$.

The main idea of our algorithm is to search primarily in $\ker(A^T)$. From a geometrical viewpoint, this is where the function exhibits non zero curvature. Actually, we would like the algorithm to generate directions $s$ such that $A^T s = 0$.

The difficulty is that $A$ is unknown before the optimization process starts, and needs to be approximated. We use $\nabla^2 f(x_k)$ as an approximation of $\nabla^2 f(x^*)$ as the algorithm proceeds. Consequently, performing an eigen-structure analysis of $\nabla^2 f(x_k)$ enables to generate the desired approximation of $A$. The eigen-subspace associated with eigenvalues of $\nabla^2 f(x_k)$ which are close to zero is used as an approximation for the range of $A$. The quality of such an approximation improves as $x_k$ converges to $x^*$.

The computational burden of a full eigen-structure analysis per iteration is often unacceptable. For example, applying the full QR-algorithm for the symmetric eigenvalue problem to $\nabla^2 f(x_k)$ would require to compute a full QR-factorization of $\nabla^2 f(x_k)$ at each iteration of the identification procedure, that is $\mathcal{O}(n^3)$ flops.

Consequently, we propose a generalization of the inverse iteration method to identify the relevant subspace. The inverse iteration is an iterative process identifying the eigenvalue of a symmetric matrix $H \in \mathbb{R}^{n\times n}$ closest (in modulus) to a given target, as well as the associated eigenvector (see, for instance, Golub and Van Loan, 1996). The method proposed in this chapter generalizes this procedure and allows to compute higher-dimensional invariant subspaces. Given a symmetric matrix $H \in \mathbb{R}^{n\times n}$, $r$ such that $1 \leq r \leq n$, and a target $\lambda$, the generalized inverse iteration considers $\bar{H} = (H - \lambda I_{n\times n})^{-1}$ and generates a matrix $\tilde{A} \in \mathbb{R}^{n\times r}$, such that $\mathrm{Im}(\tilde{A})$ approximates the dominant invariant subspace of dimension $r$ of $\bar{H}$, which is the subspace associated with the $r$ eigenvalues of $H$ which are closest (in modulus) to the given target $\lambda$.

The main steps of the generalized inverse iteration at iteration $k$ of the optimization algorithm can be summarized as follows.

- Consider an initial approximation $Q_k = Q_{\text{init}} \in \mathbb{R}^{n\times r}$ of $A$. It can be either the $r$ first columns of the identity matrix of dimension $n$, $I_{n\times n}$, or the approximation $Q_{k-1}$ obtained by the procedure at the previous iteration of the optimization algorithm.

- Repeat:

  1. Compute $Z \in \mathbb{R}^{n\times r}$ by solving

  $$(H - \lambda I_{n\times n})Z = Q_k$$

  2. Compute the new approximation $Q_k$ by performing a partial QR-factorization of $Z$, that is:

  $$Q_k R = Z$$

  until a given stopping criterion is satisfied.

Note that the partial QR-factorization is applied to a matrix belonging to $\mathbb{R}^{n\times r}$ so that $Q_k$ is only composed of $r$ columns. In comparison, a full QR-algorithm would compute at each iteration a full QR-factorization with $Q \in \mathbb{R}^{n\times n}$. The cost of this generalized inverse iteration is $\mathcal{O}(rn^2)$, which is interesting when $r$ is small compared to $n$.

The stopping criterion is based on the difference in $\ell_2$-norm between two consecutive $Q_k$ approximations. As soon as this difference is below a given threshold (typically $10^{-6}$), we stop the procedure. The last $Q_k$ approximation represents the desired approximation of $A$. We obtained the eigenvalues associated with this eigen subspace by computing

$$\lambda_i = \frac{q_i^T H q_i}{q_i^T q_i}$$

for $i = 1, \ldots, r$ where $q_i$ is the $i$-th column of $Q_k$.

In our case, we apply this method with $H = \nabla^2 f(x_k)$ and would like to identify the $\bar{r}$ eigenvectors corresponding to null eigenvalues. The dimension $\bar{r}$ is not known in advance. At each iteration $k$ of the optimization algorithm, we use the following procedure to identify the dimension of the singularity $\bar{r}$. In order to reduce the computational cost (that is, the number of times we apply the generalized inverse iteration), we make use of the value found for $\bar{r}$ at the previous iteration, which we denote $r_{\text{previous}}$.

Initialization $r_{\text{previous}} = 0$ and singular $= 0$ for the first iteration of the optimization algorithm (for the subsequent iterations, these values are determined by this procedure).

Phase 1 If singular $= 0$

- Apply the generalized inverse iteration with $r = 1$.
- If the obtained eigenvector corresponds to a non-zero eigenvalue (that is, if the problem is not declared to be singular), $\bar{r} = 0$. Set singular $= 0$, $r_{\text{previous}} = 0$ and STOP.
- If the corresponding eigenvalue is declared to be zero (according to the threshold), set singular $= 1$, $r_{\text{previous}} = 1$ and go to Phase 2.

Phase 2

- Apply the generalized inverse iteration with $r = \max(r_{\text{previous}}, 1)$.
- If all corresponding eigenvalues are close to zero, go to Phase 3a.
  (We apply the generalized inverse iteration with increasing values of $r$).
- If at least one corresponding eigenvalue is declared to be non-zero:
  If $r = 1$, $\bar{r} = 0$. Set $r_{\text{previous}} = \bar{r}$ and STOP.
  Otherwise go to Phase 3b.
  (We apply the generalized inverse iteration with decreasing values of $r$).

Phase 3a `for` $r = \max(r_{\text{previous}}, 1) + 1 : n$

- Apply the generalized inverse iteration with $r$
- If the additional eigenvalue is close to zero, continue.
- If the additional eigenvalue is non-zero, $\bar{r} = r - 1$. Set $r_{\text{previous}} = \bar{r}$ and STOP.

Phase 3b `for` $r = r_{\text{previous}} - 1 : -1 : 1$

- Apply the generalized inverse iteration with $r$
- If it remains at least one non-zero eigenvalue, continue. If $r = 1$, $\bar{r} = 0$. Set $r_{\text{previous}} = 0$ and STOP.
- If all obtained eigenvalues are close to zero, $\bar{r} = r$. Set $r_{\text{previous}} = \bar{r}$ and STOP.

As $r$ is usually small compared to $n$ and does not change too much from iteration to iteration of the optimization algorithm, the cost of this procedure using the generalized inverse iteration is significantly lower than the one of a full QR-analysis. Moreover, this allows us to compute only relevant information for our purposes.

Note that the generalized inverse iteration fails with $\lambda = 0$ and we have to use a small positive value as target, such as $\lambda = 10^{-10}$. Also, we declare an eigenvalue to be null if its absolute value is less than $10^{-6}$.

Now that the singularity is identified, we need to use this information to help the optimization algorithm. The central idea described in the next section is to constrain directions to lie in the subspace in which we have relevant information about curvature by using a penalty approach.

## 2.5 Trust-region based algorithms

In this chapter, we focus on trust-region based methods. Indeed, these methods present significant theoretical and practical advantages, and can easily be adapted with many variants (see Conn et al., 2000). We start by presenting the classical trust-region framework for an optimization algorithm dedicated to solve unconstrained nonlinear optimization problems. An iteration $k$ of a trust-region based algorithm can be summarized by the following steps:

**Step 1: Model definition.** Define a quadratic model $m_k$ (typically using a truncated Taylor's series) of the objective function in a region $\mathcal{B}_k$ (called the *trust-region*) where this model can be trusted.

**Step 2: Step computation.** Compute a step $s_k$ that sufficiently reduces the model $m_k$ and such that $x_k + s_k \in \mathcal{B}_k$. This step is also called the trust-region subproblem because we approximately solve the following problem

$$\begin{cases} \min m_k(x_k + s) \\ \text{s.t. } x_k + s \in \mathcal{B}_k, \end{cases}$$

that is, minimizing the model within the trust-region.

**Step 3: Acceptation of the trial point.** Assess the quality of the trial step $s_k$ and decide whether $x_k + s_k$ is accepted as the next iterate $x_{k+1}$ or not.

**Step 4: Trust-region radius update.** Update the size of the trust-region.

Minimizing the quadratic model under the trust-region constraint is the core of the algorithm. Many methods have been proposed in the literature, such as "dogleg" or truncated conjugate-gradient (see Conn et al., 2000 for a review). In the latter case, preconditioning techniques have shown to improve the numerical behavior of the algorithm for difficult problems, such as the modified Cholesky factorization by Schnabel and Eskow (1999), available in the LANCELOT package (Conn et al., 1992).

The assessment of the model's quality is performed in general by comparing the improvement predicted by the model with the actual improvement of the objective function. Advanced techniques inspired from multi-criteria optimization have recently emerged, exhibiting faster convergence. Originally proposed by Fletcher and Leyffer (2002), these techniques are called "filter" methods.

Now we present different variants of this general scheme. Variants A and C are from the literature. Variants B and D are new ideas proposed in this chapter.

### 2.5.1 Variant A: A trust-region algorithm

We first propose to use the basic trust-region algorithm, as described in Conn et al. (2000). In this variant, we consider the following specific steps:

**Step 1a: Model definition.** Define $m_k$ in $\mathcal{B}_k$ (where $\mathcal{B}_k$ is a sphere centered at $x_k$ of radius $\Delta_k$) as a quadratic model of f around $x_k$, that is:

$$m_k(x_k + s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s \tag{2.2}$$

**Step 2a: Step computation.** The original trust-region subproblem is defined as

$$\begin{cases} \min m_k(x_k + s) \\ \text{s.t. } \|s\| \leq \Delta_k, \end{cases} \tag{2.3}$$

where $\Delta_k$ is the radius of the trust-region.

**Step 3a: Acceptation of the trial point.** Compute $f(x_k + s_k)$ and define

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

If $\rho_k \geq \eta_1$, then define $x_{k+1} = x_k + s_k$; otherwise define $x_{k+1} = x_k$.

### 2.5.2 Variant B: A new trust-region algorithm

We propose a new trust-region algorithm to deal with singularity. It is an extension of Variant A where the trust-region subproblem is modified, involving the matrix $Q_k$ defined in Section 2.4.

To achieve our objective of generating directions $s$ such that $A^T s = 0$, we propose to penalize directions $s$ such that $\|Q_k^T s\| > 0$, by modifying the model of the objective function as well as the trust-region subproblem. We consider the following specific steps:

**Step 1b: Model definition.** Define $\widehat{m}_k$ as follows:

$$\widehat{m}_k(x_k + s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s + \frac{1}{2} c \|Q_k^T s\|^2 \tag{2.4}$$

**Step 2b: Step computation.** The corresponding trust-region subproblem is defined as

$$\left\{ \begin{array}{l} \min \widehat{m}_k(x_k + s) = m_k(x_k + s) + \frac{1}{2} c \|Q_k^T s\|^2 \\ \text{s.t. } \|s\| \leq \Delta_k, \end{array} \right. \tag{2.5}$$

where $c \geq 0$ is the penalty parameter.

**Step 3b: Acceptation of the trial point.** Identical to Variant A.

We set $c = 0$ if $\nabla^2 f(x_k)$ is detected to be nonsingular. The second derivatives matrix of the new model is given by

$$\nabla^2 \widehat{m}_k(x_k) = \nabla^2 f(x_k) + c\, Q_k Q_k^T. \tag{2.6}$$

It means that we add a multiple of the $Q_k Q_k^T$ matrix to the second derivatives matrix of $f$ when it is close to singularity. Geometrically, it amounts to "bending" the function in the subspace where there is originally no curvature. More precisely, eigenvalues of $\nabla^2 f(x_k)$ close to 0 take the value $c > 0$ in $\nabla^2 \widehat{m}_k(x_k)$.

The penalty parameter $c$ is chosen as small as possible so that the perturbation of the model is not too severe. In practice, we start with $c = 1$, and test if the direction $s^*$, solution of (2.5), is such that $\|Q_k^T s^*\|$ is sufficiently close to zero (typically, $\|Q_k^T s^*\| \leq 10^{-3}$). If not, $c$ is multiplied by 10 for the next iteration, until it reaches the upper bound $\kappa_c$ (typically $10^5$).

In addition to the obvious numerical reasons, this upper bound allows the new model to satisfy the general assumptions of the trust-region framework, in particular the fact that all eigenvalues of the second derivative matrix of the model must stay bounded. Consequently, convergence to a first-order critical point of the optimization problem can be guaranteed.

According to Conn et al. (2000), the trust-region based algorithm described above converges to first-order critical points if the following assumptions on the model are valid:

A.M.1 For all $k$, the model $\widehat{m}_k$ is twice differentiable.

A.M.2 The values of the objective function and of the model coincide at the current iterate; that is, for all $k$

$$\widehat{m}_k(x_k) = f(x_k).$$

A.M.3 The values of the gradient of the objective function and of the gradient of the model coincide at the current iterate; that is, for all $k$

$$\nabla \widehat{m}_k(x_k) = \nabla f(x_k).$$

A.M.4 The Hessian of the model remains bounded within the trust-region; that is,

$$\|\nabla^2 \widehat{m}_k(x_k)\| \leq \kappa_{umh} - 1 \text{ for all } x \ \in \mathcal{B}_k,$$

for all $k$, where $\kappa_{umh} \geq 1$ is a constant independent of $k$.

We briefly prove that the model $\widehat{m}_k$ satisfies these assumptions. To do this we first compute the first and second-order derivatives of $\widehat{m}_k$ which gives:

$$\nabla \widehat{m}_k(x_k + s) = \nabla f(x_k) + \nabla^2 f(x_k)^T s + c\, Q_k Q_k^T s, \tag{2.7}$$

and

$$\nabla^2 \widehat{m}_k(x_k + s) = \nabla^2 f(x_k) + c\, Q_k Q_k^T. \tag{2.8}$$

Using (2.7) and (2.8) and the assumption that the objective function $f$ is twice differentiable, we directly obtain A.M.1. A.M.2 results from (2.4). Taking $s = 0$ in (2.7) gives immediately A.M.3. A.M.4 remains to be proved.

From (2.8), we have that:

$$\|\nabla^2 \widehat{m}_k(x_k)\| \leq \|\nabla^2 f(x_k)\| + c\|Q_k Q_k^T\| \leq \kappa_{ufh} + c \tag{2.9}$$

by using assumptions on $f$ (namely the boundedness of the Hessian matrix) and the fact that columns of the matrix $Q_k$ generated by the identification procedure have norm 1.

We can conclude as we put an upper bound $\kappa_c$ on the value of the penalty parameter $c$. Thus there exists a constant $\kappa_{umh} \geq 1$ such that

$$\|\nabla^2 \widehat{m}_k(x_k)\| \leq \kappa_{umh} - 1 \tag{2.10}$$

for all $k$. It is sufficient to take $\kappa_{umh} \geq \kappa_{ufh} + c + 1$. The constant being independent from $k$, we have the uniform boundedness.

### 2.5.3 Variant C: A standard filter algorithm

The concept of the filter has been introduced in nonlinear optimization by Fletcher and Leyffer (1997) (and consequently published as Fletcher and Leyffer, 2002) and Fletcher et al. (2002). Inspired from multi-criteria optimization, it provides a great deal of flexibility to measure progress toward the solution of a problem, both in terms of optimality and feasibility. The authors motivated the use of the filter in a trust-region Sequential Quadratic Programming (SQP) framework by the claim that it provides a global optimization safeguard that interferes as little as possible with Newton's method. Fletcher and Leyffer (2002) define a 2-dimensional filter associated with the two objectives of constrained optimization, namely minimizing the objective function while satisfying the constraints. Gould et al. (2005) generalize the concept by using a multidimensional filter to solve systems of nonlinear equations as well as nonlinear least-squares. A multidimensional filter is also used in Gould et al. (2006) in the context of unconstrained optimization. The advantage of the filter is the increased flexibility in the optimization algorithm to accept new and larger iterates and to take full step more often compared to the classical trust-region scheme, and consequently, a potentially faster convergence.

Our third algorithm is an adaptation of the algorithm proposed by Gould et al. (2006), with the following two modifications:

1. the flag RESTRICT is never set;
2. the test to accept the trial step (step 3) has been modified.

The first two steps of this variant are the same as Variant A, that is we used the classic model (2.2) and the original trust-region subproblem (2.3). The specific feature of this variant is the test for acceptance of the trial point $x_k^+ = x_k + s_k$.

We extend the trust-region algorithm (Variant A) by introducing a multidimensional filter technique, whose aim is to encourage the convergence of iterates to a first-order critical point, by driving each component of the gradient of the objective function $\nabla f(x) = g(x) = (g_1(x), \ldots, g_n(x))^T$ to zero.

The notion of filter is based on the concept of dominance. In our case, we say that an iterate $x_1$ is dominated by an iterate $x_2$ when

$$|g_i(x_2)| \leq |g_i(x_1)| \quad \forall i = 1, \ldots, n.$$

Consequently, we consider that the iterate $x_1$ is of no interest if we keep the iterate $x_2$. Indeed, $x_2$ is better than $x_1$ with regard to each component of the gradient.

Given this concept, we remember all non-dominated iterates by using the filter structure. We define the multidimensional filter $\mathcal{F}$ as the list of $n$-tuples $(g_{k,1}, \ldots, g_{k,n})$ with $g_{k,i} = g_i(x_k)$ such that, if $g_k \in \mathcal{F}$, then we have that

$$|g_{k,j}| < |g_{l,j}| \text{ for at least one } j \in \{1, \ldots, n\}$$

$\forall\ g_l \in \mathcal{F}$. It means that each point in the filter is not dominated by any other point in the filter.

In a filter method, we accept a new trial point $x_k^+$ if it is not dominated by any other point in the filter. However, from an algorithmic point of view, we do not want to accept a trial point which is arbitrarily close to a point in the filter. This is why we slightly strengthen the acceptability test and we thus say that a trial point $x_k^+$ is acceptable for the filter $\mathcal{F}$ if

$$\forall\ g_l \in \mathcal{F} \quad \exists\ j \in \{1, \ldots, n\} \text{ such that } |g_j(x_k^+)| \leq (1 - \gamma_\theta)|g_{l,j}|,$$

where $\gamma_\theta$ is a small positive constant. If an iterate $x_k$ is acceptable for the filter and if we decide to add it to the filter, we remove all dominated entries $g_l \in \mathcal{F}$ such that $|g_{l,j}| > |g_{k,j}|\ \forall\ j \in \{1, \ldots, n\}$.

As the presented filter mechanism only guides the iterates toward a zero gradient, it is adequate for convex problems where a zero gradient is both necessary and sufficient condition for second-order optimality but it may be inappropriate for nonconvex ones. We thus adapt the above mechanism by a reset to zero of the filter after an iteration for which a sufficient decrease in the objective function is achieved with the model $m_k$ being nonconvex. In this case, we also define an upper bound on the acceptable objective function values in order to keep a monotone algorithm in term of objective function value.

The specific steps of this variant are the following:

**Step 1c: Model definition.** Identical to Variant A.

**Step 2c: Step computation.** Identical to Variant A.

**Step 3c: Acceptation of the trial point.**

- If $x_k^+$ is acceptable for the filter $\mathcal{F}$ and $\texttt{nonconvex}$[1] is unset
  Set $x_{k+1} = x_k^+$ and add $g_k^+$ to the filter $\mathcal{F}$ if $\rho_k < \eta_1$.
- If $x_k^+$ is not acceptable for the filter $\mathcal{F}$ or $\texttt{nonconvex}$ is set
  If $\rho_k \geq \eta_1$ then
    Set $x_{k+1} = x_k^+$ and, if $\texttt{nonconvex}$ is set, set $f_{\text{sup}} = f(x_{k+1})$
    and reinitialize the filter $\mathcal{F}$ to the empty set;
  else Set $x_{k+1} = x_k$.

This filter variant accepts more often the trial point than the original trust-region algorithm. Indeed, if the trial point is acceptable for the filter, we move toward this point and if it is not, we look at the quality of the reduction factor $\rho_k$ as in the first algorithm. Note that an iteration of this filter method is equivalent to a basic trust-region iteration when the function is nonconvex. The idea is to let the filter play the major role while convexity is encountered and fall back to the classical trust-region framework if non-convexity is detected.

### 2.5.4 Variant D: A new filter algorithm

We now consider a new filter algorithm to deal with singularity based on variant C exactly in the same way that we derived Variant B from Variant A. We consider the following specific steps:

**Step 1d: Model definition.** Define $\widehat{m}_k$ as follows:

$$\widehat{m}_k(x_k + s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s + \frac{1}{2} c \|Q_k^T s\|^2 \tag{2.11}$$

**Step 2d: Step computation.** The corresponding trust-region subproblem is defined as

$$\begin{cases} \min \widehat{m}_k(x_k + s) \\ \text{s.t. } \|s\| \leq \Delta_k, \end{cases} \tag{2.12}$$

where $c \geq 0$ is the penalty parameter.

**Step 3d: Acceptation of the trial point.** Identical to Variant C.

Following the convergence theory in Gould et al. (2006), the new model we propose in this filter variant must satisfy a major assumption in order to guarantee that the sequence of iterates produced by the filter algorithm converges to first-order critical points. More precisely, for all $k$, the model

$$\widehat{m}_k(x_k + s) = m_k(x_k + s) + \frac{1}{2} c \|Q_k^T s\|^2$$

[1] see Gould et al., 2006 for details

has to be twice differentiable on $\mathbb{R}^n$ and must have a uniformly bounded Hessian.

Firstly, it is obvious to prove the twice differentiability (see (2.7) and (2.8)). Secondly, the uniform boundedness is obtained directly from (2.9) and (2.10) as this new filter algorithm makes use of the same model as Variant B corresponding to the new trust-region algorithm.

We also consider preconditioned versions of variants A and C, denoted $A_p$ and $C_p$. As preconditioning matrix, we use a modified Cholesky factorization of the second derivatives matrix $\nabla^2 f(x_k)$. More precisely, the preconditioner is obtained following the lines of Schnabel and Eskow (1999).

To summarize, we consider a total of 6 algorithms, namely:

- The trust-region algorithm presented in Section 2.5.2 (Variant B) and the filter-trust-region algorithm presented in Section 2.5.4 (Variant D) both designed to handle singularity by the means of the perturbed trust-region subproblem (2.5) and the procedure described in Section 2.4.

- The basic trust-region algorithm (Variant A) and an adaptation of the standard filter-trust-region method (Variant C) using the classical model of the objective function (2.2).

- The preconditioned versions of Variant A and Variant C, $A_p$ and $C_p$.

### 2.5.5 Implementation issues

- In practical tests, the trust-region subproblem consists in minimizing model (2.2) subject to the trust-region constraint, except that we approximate the second order derivatives matrix at the current iterate $x_k$, that is $\nabla^2 f(x_k)$, by a matrix $H_k$ obtained using finite differences.

- The trust-region subproblem for the four first algorithmic variants is solved using a Truncated Conjugate Gradient method (see Toint, 1981, Steihaug, 1983 or Conn et al., 2000).

- For Variants $A_p$ and $C_p$, we use a preconditioned conjugate gradient framework (see, for instance, Conn et al., 2000) instead of the standard conjugate gradient algorithm for solving the trust-region subproblem (2.3).

## 2.6 Numerical experiments

In this section, we present an analysis of the performances of new algorithmic variants compared to classical trust-region and filter algorithms from the literature. Section 2.6.1 contains a description of the set of test problems which have been used for the numerical experiments. The methodology for performance analysis is described Section 2.6.2. Sections 2.6.3-2.6.6 present results on singular problems while Section 2.6.7 shows the performance of proposed algorithms on non-singular problems.

### 2.6.1 Description of test problems

The set of test functions has been proposed by Moré et al. (1981). It is composed, among other things, of 34 unconstrained optimization problems. Most of these problems have a non-singular second derivatives matrix at the local minimum. As we want to perform tests on singular problems, we use the technique proposed by Schnabel and Frank (1984) to modify the problems of Moré et al. (1981) and create singular optimization problems such that the second derivatives matrix has a rank $n - k$ at the local minimum where $n$ is the dimension of the problem and $1 \leq k \leq n$ is the dimension of the singularity. In this chapter we focus on problems having a second-order derivatives matrix of rank $n - 1$ or $n - 2$ at the local minimum as in Schnabel and Chow (1991). Tests have been actually performed on 38 problems containing a singularity of dimension 1 (that is one null eigenvalue) at the local minimum:

- 29 problems with dimension between 2 and 11,
- 3 problems with a dimension $n$ which can be parametrized. In this case, we have used $n = 10, 20, 40$.

We also carried out tests on a set of 38 test functions whose second derivatives matrix has rank $n - 2$ at $x^*$, namely:

- 29 problems with dimension between 3 and 11,
- 3 problems with a dimension $n$ which can be parametrized. In this case, we have used $n = 10, 20, 40$.

For each problem, we have used the starting point given in the original paper of Moré et al. (1981).

Note that all tested algorithms have converged to the same solution for all 76 problems (when they did not fail to converge). Moreover, this solution corresponds to the local minimum at which a given problem is singular.

To summarize, we thus have a set of 76 test problems in which the singularity has been explicitly incorporated.

### 2.6.2 Performance analysis

We present in the next sections a performance analysis of the variants presented in Section 2.5. All algorithms and test functions have been implemented with the package Octave (see `www.octave.org` or Eaton, 1997b) and computations have been done on a desktop equipped with 3GHz CPU, in double precision.

The stopping criterion for all algorithms is a composition of two conditions: gradient close to zero, that is $\|\nabla f(x_k)\| \leq 10^{-6}$, and maximum number of iterations fixed to 1000. The measure of performance is the number of iterations or the CPU time necessary to reach convergence (as defined above). We are presenting the results following the performance profiles analysis method proposed by Dolan and Moré (2002). If $f_{p,a}$ is the performance index (the number of function evaluations, or the CPU time) of algorithm $a$ on problem $p$, then the *performance ratio* is defined by

$$r_{p,a} = \frac{f_{p,a}}{\min_b\{f_{p,b}\}}, \tag{2.13}$$

if algorithm $a$ has converged for problem $p$, and $r_{p,a} = r_{\text{fail}}$ otherwise, where $r_{\text{fail}}$ must be strictly larger than any performance ratio (2.13) corresponding to a success. For any given threshold $\pi$, the overall performance of algorithm $a$ is given by

$$\rho_a(\pi) = \frac{1}{n_p}\Phi_a(\pi) \tag{2.14}$$

where $n_p$ is the number of problems considered, and $\Phi_a(\pi)$ is the number of problems for which $r_{p,a} \leq \pi$. In particular, the value $\rho_a(1)$ gives the proportion of times that algorithm $a$ wins over all other algorithms. The value $\rho_a(\pi)$ with $\pi \geq r_{\text{fail}}$ gives the proportion of times that algorithm $a$ solves a problem and, consequently, provides a measure of the robustness of each method.

Note that the sum of $\rho_a(1)$ values for all algorithms $a$ considered in a given profile may exceed 1 in the case that some algorithms perform exactly the same on some of the tested problems.

### 2.6.3 TR and filter methods

We first compare variants A to D. Figure 2.1 represents the full profile while Figure 2.2 provides a zoom on $\pi$ between 1 and 3. In terms of number of iterations, we can see that the two best algorithms are the new variants B and D. These new algorithms significantly outperform the classical ones both in efficiency and robustness. Note also that the new filter algorithm (Variant D) outperforms the new trust-region (Variant B) algorithm. Similarly, the standard filter method (Variant C) shows a better efficiency than the basic trust-region method (Variant A), consistently with the findings of Gould et al. (2006). Note also that filter variants are more robust than trust-region variants as they are able to solve all 76 problems on which algorithms have been tested.

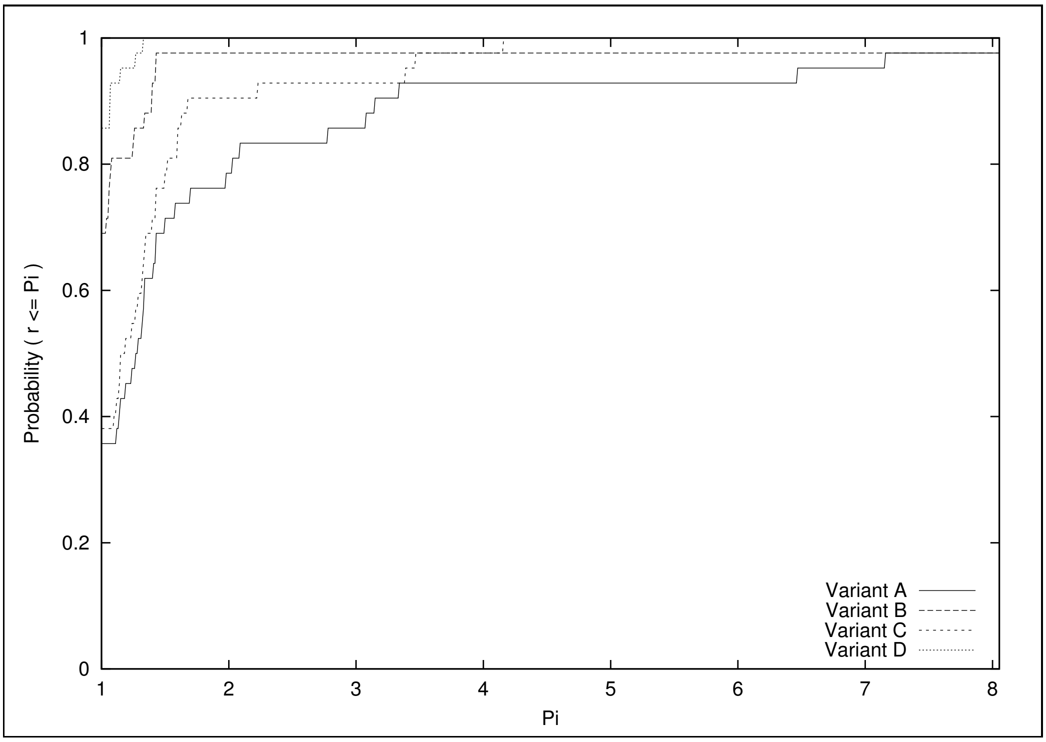


Figure 2.1: Comparison of the number of iterations for Variants A,B,C,D

Figures 2.3 and 2.4 show the performance of the same variants in terms of CPU time. From Figure 2.4, we can already see that there is a computational overhead associated with the new variants proposed in this chapter. It is mainly due to the computational cost of the identification procedure described in Section 2.4. It is easy to measure this overhead on specific profiles for trust-region and filter variants presented in the next subsections. We can also easily see that filter methods compensate the numerical algebra associated with the management of the filter by a higher efficiency compared to trust-region algorithms on which they are based.

### 2.6.4 TR methods

We now compare variants A and B in Figure 2.5(a). Figure 2.5(b) provides a zoom on $\pi$ between 1 and 3. The performance criterion is the number of iterations to reach

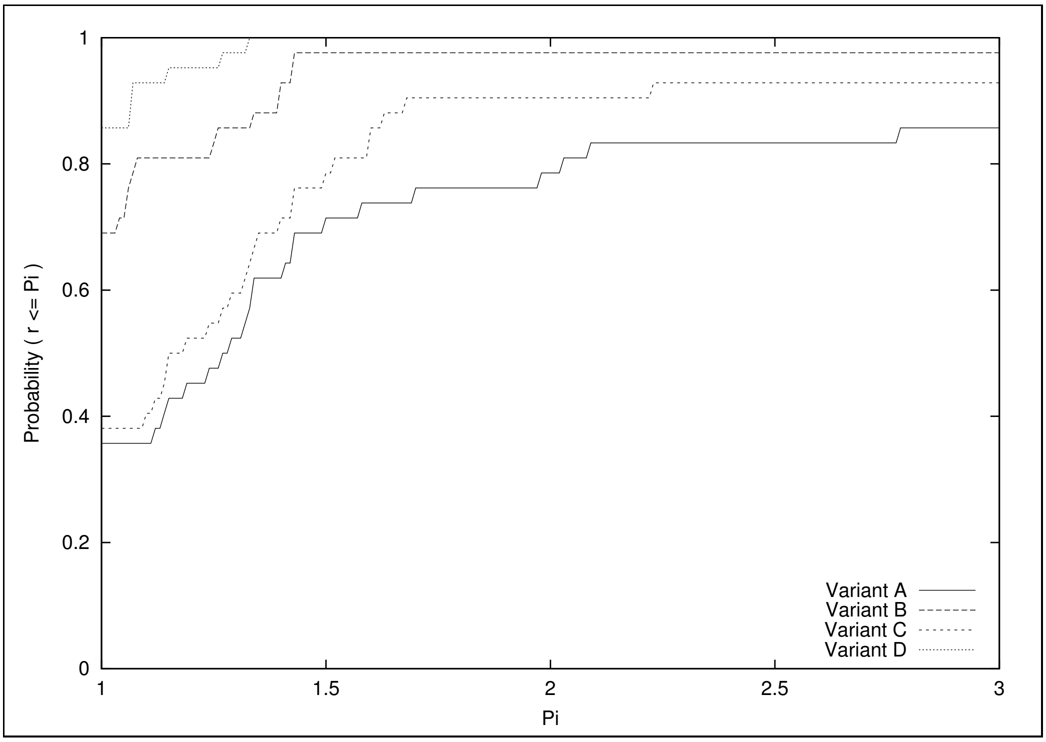


Figure 2.2: Zoom on the number of iterations for Variants A,B,C,D

convergence. Variant B performs significantly better than the classical algorithm both in terms of efficiency and robustness. From Figure 2.5(b), we see that it is the best on 90% of the 76 singular problems tested. When it is not the best algorithm, it converges within a factor around 1.25 of the classical trust-region algorithm on all 76 tested problems.

In Figures 2.6(a) and 2.6(b), we compare variants A and B with regard to the CPU time. Our variant B is still the best method with regard to this measure of performance, even if we can see from these profiles that there is a computational overhead. As we already mentioned, it is mainly due to the numerical algebra of the identification procedure, that is the procedure described in Section 2.4 based on the generalized inverse iteration. Indeed the difference between the profiles associated with the competitors is smaller than previously. However, it is important to note that, even if the test problems do not have an objective function expensive to compute, the higher efficiency of the new variant compensates its computional overhead. Despite the additional effort in computation due to the singularity identification process, we see that the new algorithm takes, on more than 60% of the problems, less time to reach convergence thanks to the smaller number of iterations necessary to converge to a local minimum. On some problems, the new algorithm is up to 5 times faster than the standard one in terms of computational time. It is an indication

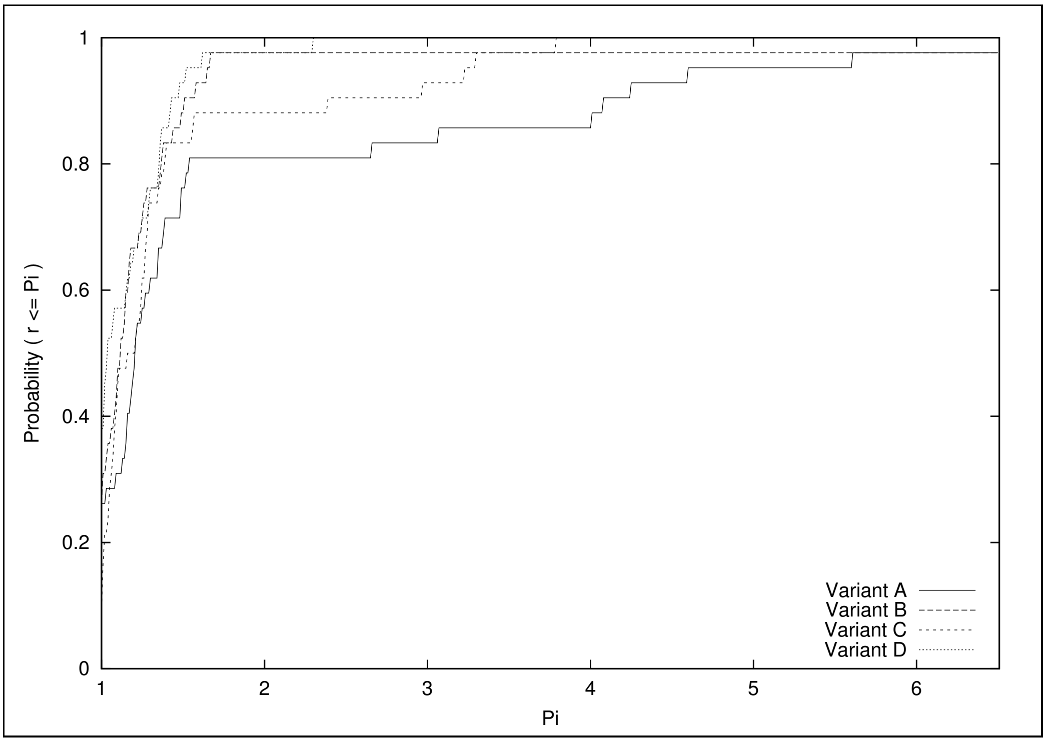


Figure 2.3: Comparison of the CPU time for Variants A,B,C,D

that the new method is particularly appropriate when the function is computationally expensive to compute.

### 2.6.5 Filter methods

Here we compare the standard filter method (Variant C) with the variant proposed in Section 2.5.4 (Variant D). Figure 2.7(a) represents the full profile while Figure 2.7(b) provides a zoom on $\pi$ between 1 and 3. The proposed variant significantly outperforms the adaptation of the filter algorithm proposed by Gould et al. (2006) in terms of number of iterations necessary to reach the convergence criterion. The new filter algorithm is the most efficient on almost all 76 tested problems. When it is not the best algorithm, it converges within a factor close to 1 of the standard filter algorithm. Note that the methods are similar in terms of robustness.

Figures 2.8(a) and 2.8(b) show the performance of variants C and D in terms of CPU time. As it was the case when analyzing the performances of the new trust-region algorithm, we can easily observe that the computational overhead associated with the proposed filter method is compensated by its better efficiency. Our variant is the fastest algorithm in CPU time on nearly 65% of the tested problems. On some of the problems, it is up

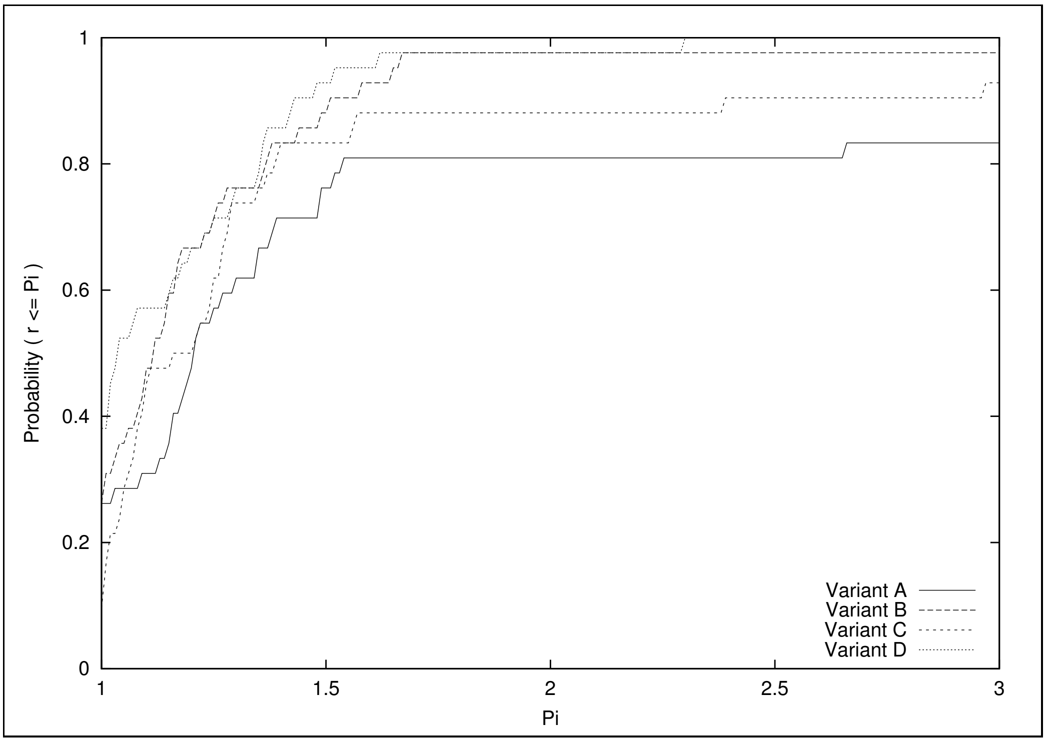


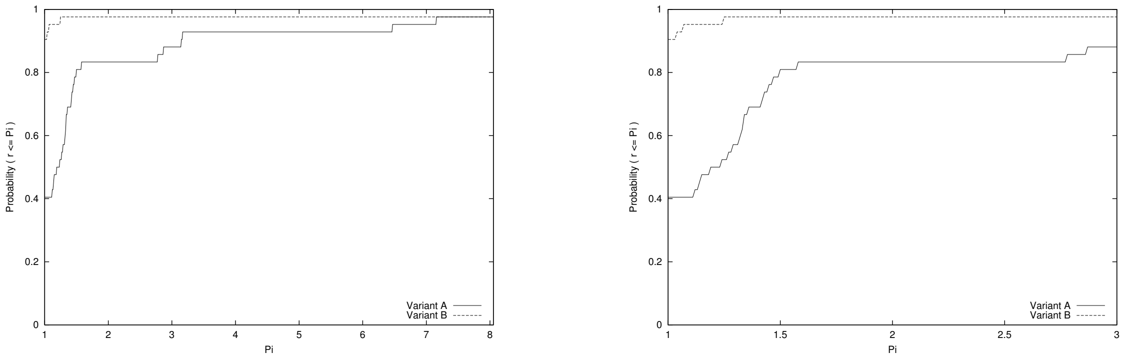
Figure 2.4: Zoom on the CPU time for Variants A,B,C,D

to 4 times faster to reach a solution. Again, we expect the advantage in CPU time to be larger for expensive functions.

### 2.6.6 Preconditioned versions vs. our variants

Here we compare preconditioned versions of trust-region (Variant $A_p$) and filter (Variant $C_p$) algorithms with variants B and D. We want to check if well-known preconditioning techniques would be a simple way of efficiently dealing with singularity issues in unconstrained optimization problems. Indeed, these techniques have shown their advantages when solving problems presenting numerical difficulties. However, we clearly see from Figures 2.9-2.12 that our variants perform significantly better than preconditioned versions of A and C.

These preconditioning techniques are not designed to deal with the type of problems we consider in the scope of this chapter. Indeed, the difficulty is due to the very small eigenvalues in the Hessian matrix of the objective function $f$. This specificity is taken into account by defining a new model of the objective function in (2.5) when a singularity is identified. As the second derivatives matrix of this model is given by (2.8), this procedure can be viewed as shifting very small eigenvalues of the Hessian matrix at the current iterate



(a) Full profile

(b) Zoom

Figure 2.5: Comparison of the number of iterations for Variants A and B

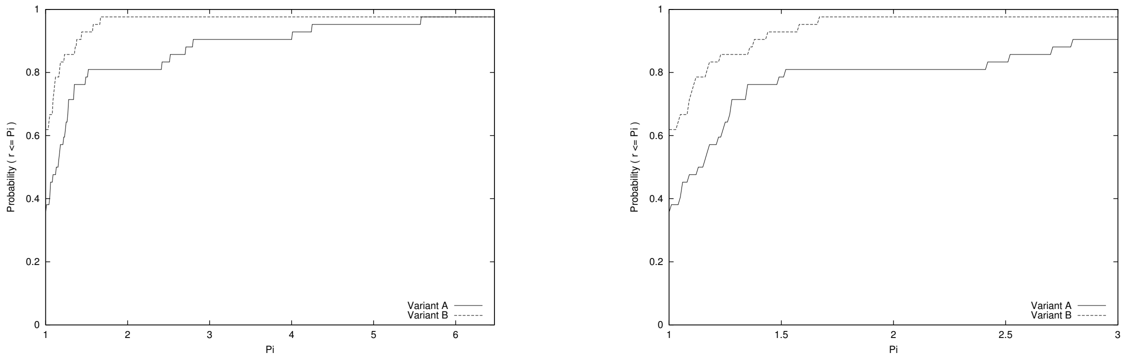
to moderate values whose magnitude is controlled by the penalty parameter c. It means that the technique we use in the proposed variants of trust-region and filter methods is acting exactly on the eigenvalues causing numerical difficulty.

### 2.6.7 Test on non-singular problems

We now present some tests on non-singular optimization problems. The idea is to analyze the computational overhead associated with the procedure described in Section 2.3 but also to see how our algorithmic variants behave on classical unconstrained optimization problems which do not exhibit singularity issues. The tests presented below have been achieved on 32 problems among the set of test functions proposed by Moré et al. (1981) which have been themselves selected from the CUTEr collection (see Gould et al., 2002).

We first compare the basic trust-region algorithm (Variant A) with the corresponding variant proposed in this chapter (Variant B). Figure 2.13 represents the profile in terms of number of iterations while Figure 2.14 provides the profile in terms of CPU time. From Figure 2.13, we can see that performances of algorithms are similar on standard problems with a slight deterioration for the new algorithm. This is not surprising in the sense that our variant basically falls back to the classical trust-region framework if no singularity has been identified during the course of the algorithm. When looking at Figure 2.14, it clearly shows the computational cost of additional numerical algebra of our variant. Indeed, profiles are closer to each other compared to the profiles of Figure 2.6(a) obtained on singular problems. Moreover, the classical trust-region algorithm is faster in terms of CPU time on more than 60% of the tested problems as expected.

Figures 2.15 and 2.16 present the performance profiles associated with both filter



(a) Full profile

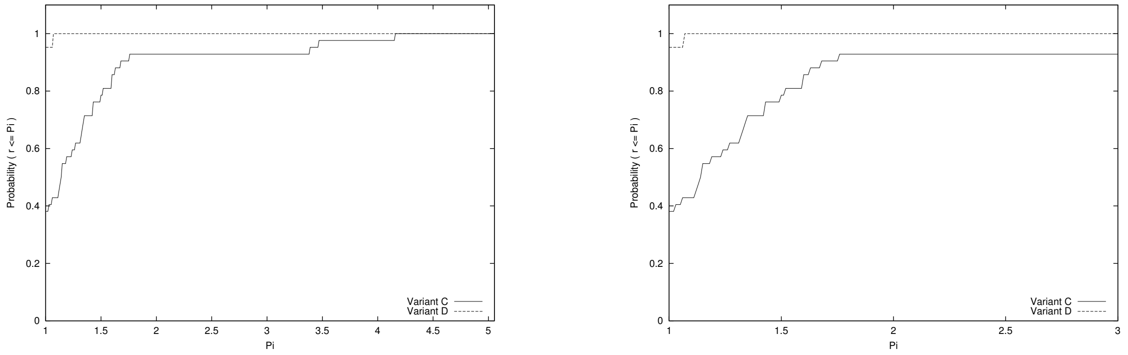
(b) Zoom

Figure 2.6: Comparison of the CPU time for Variants A and B

variants (Variants C and D) on the same 32 non-singular problems. Similarly to trust-regions algorithms tested above, filter methods exhibit the same performance in terms of efficiency and robustness, as showed by Figure 2.15. Again, we can see the impact of the overhead associated with the new filter when solving non-singular problems if we compare Figures 2.16 and 2.8(a).

## 2.7 Conclusions and perspectives

The chapter addresses with an important and difficult problem: dealing with singular problems in nonlinear optimization. It is important because it arises often in practice, especially in the early stages of a modeling process, when the models to be optimized are not completely well defined. It is difficult because the efficiency of existing algorithms is characterized by the curvature of the objective function, which is 0 (or numerically close to it) for singular problems. We have proposed a simple technique to deal with singularities. It consists in artificially adding curvature, to allow existing methods to perform decently. This requires the identification of the subspace where the function is singular, which is achieved by the generalization of a classical technique in numerical linear algebra, that is the inverse iteration method. We have shown the superiority of our approach with respect to others from the literature on a large set of problems. Namely, it appears that the computational overhead of the generalized inverse iteration method is compensated by a significant decrease in the number of iterations. This makes the method particularly appealing for problems where the CPU time spent in function evaluations is important, such as those involving simulation.

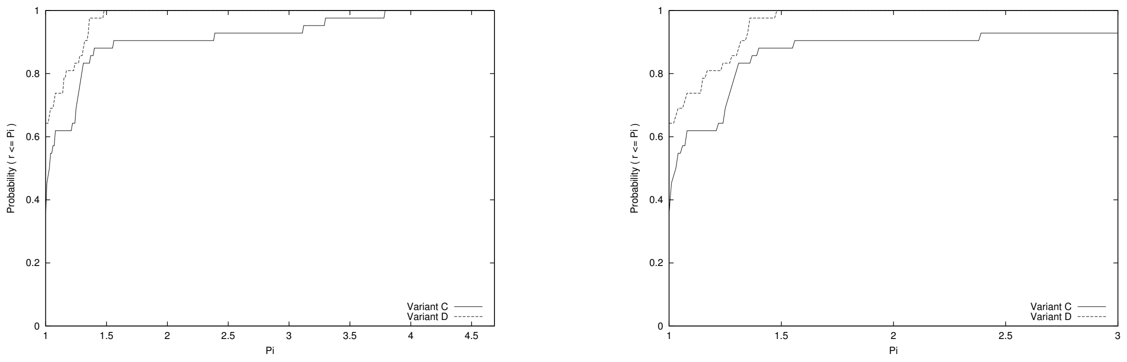


(a) Full profile

(b) Zoom

Figure 2.7: Comparison of the number of iterations for Variants C and D

No specific theoretical analysis of the convergence of the method has been performed. We have shown that the method is consistent with the general framework of trust-region methods, and inherits its convergence properties. A specific analysis of the speed of convergence is left for future work. Also, it would be natural to generalize the proposed approach to constrained problems.



(a) Full profile

(b) Zoom

Figure 2.8: Comparison of the CPU time for Variants C and D

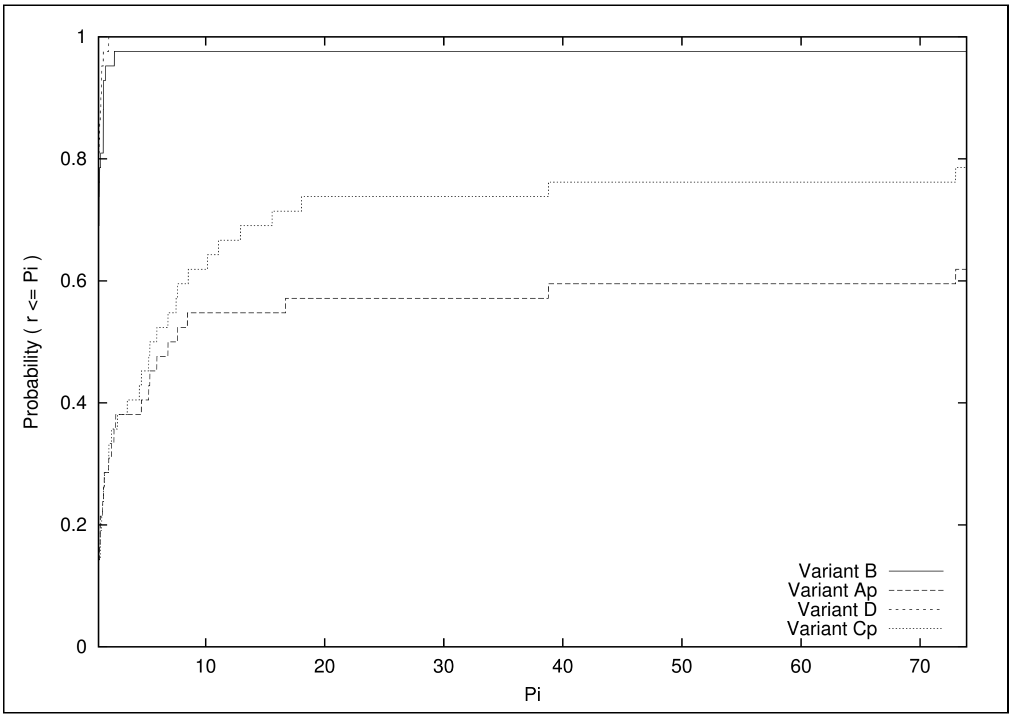


Figure 2.9: Comparison of the number of iterations for Variants B,$A_p$,D,$C_p$

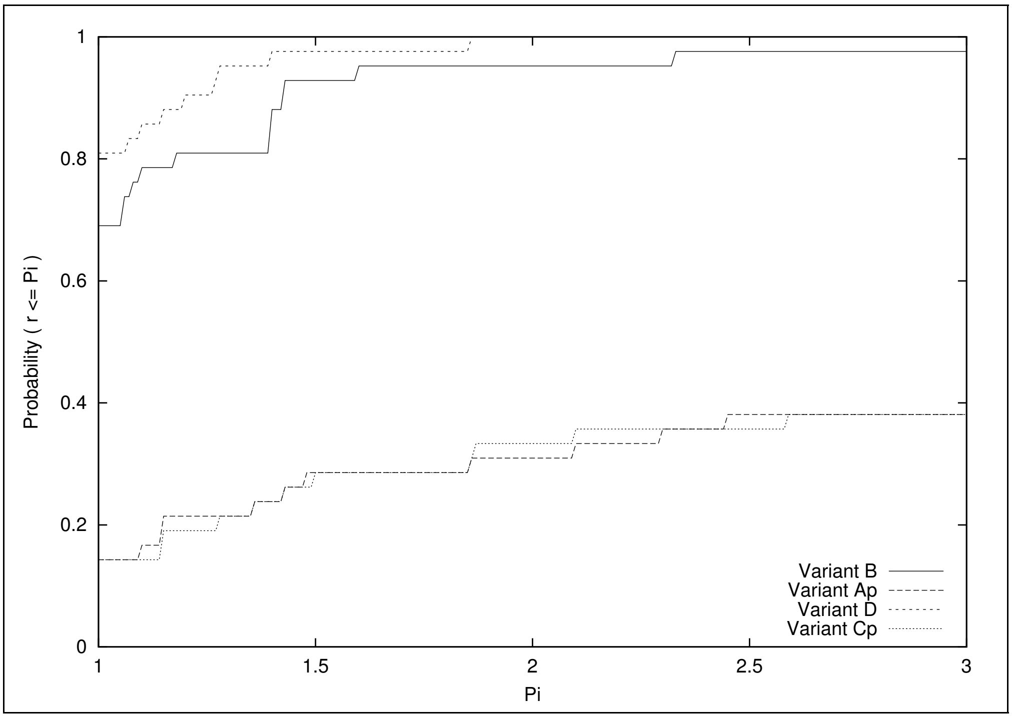


Figure 2.10: Zoom on the number of iterations for Variants B,$A_p$,D,$C_p$

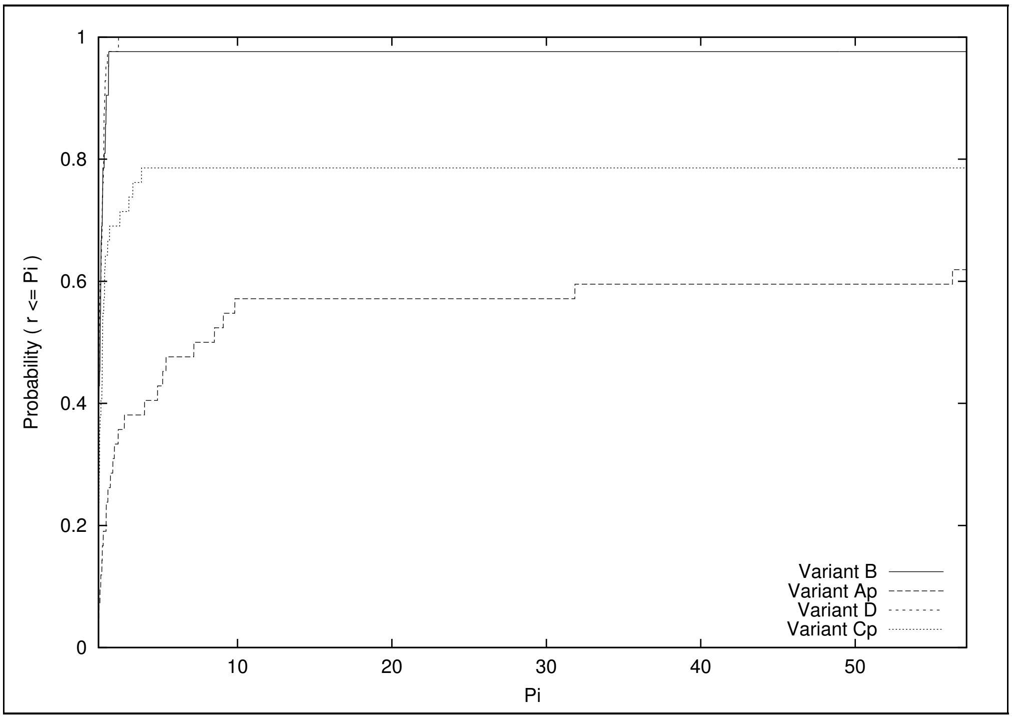


Figure 2.11: Comparison of the CPU time for Variants B,$A_p$,D,$C_p$

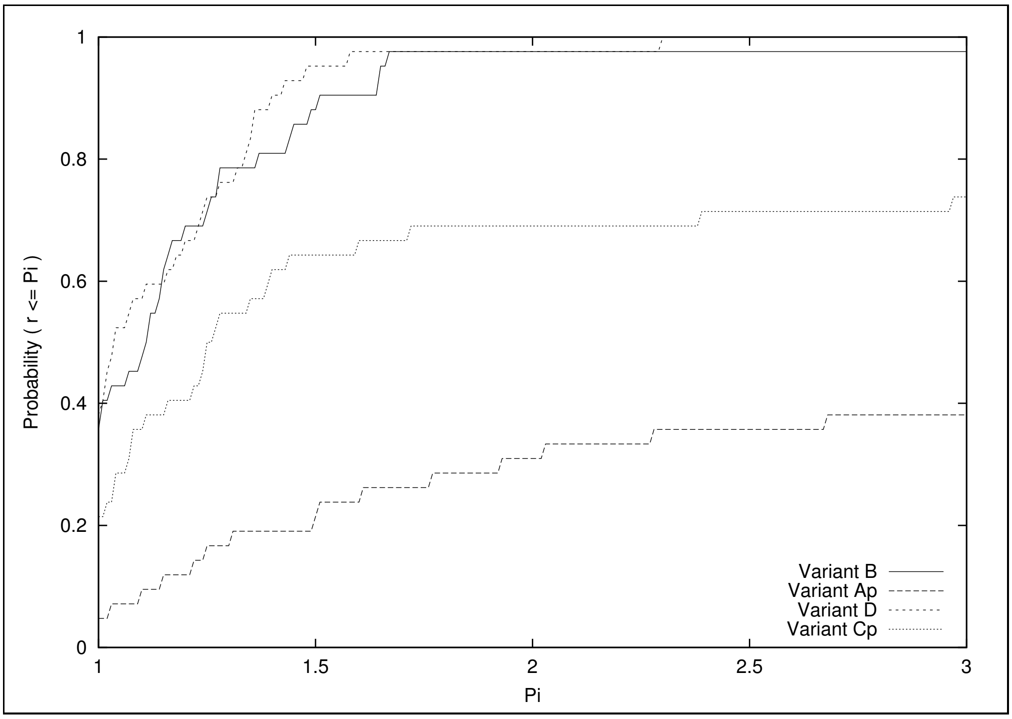


Figure 2.12: Zoom on the CPU time for Variants B,$A_p$,D,$C_p$

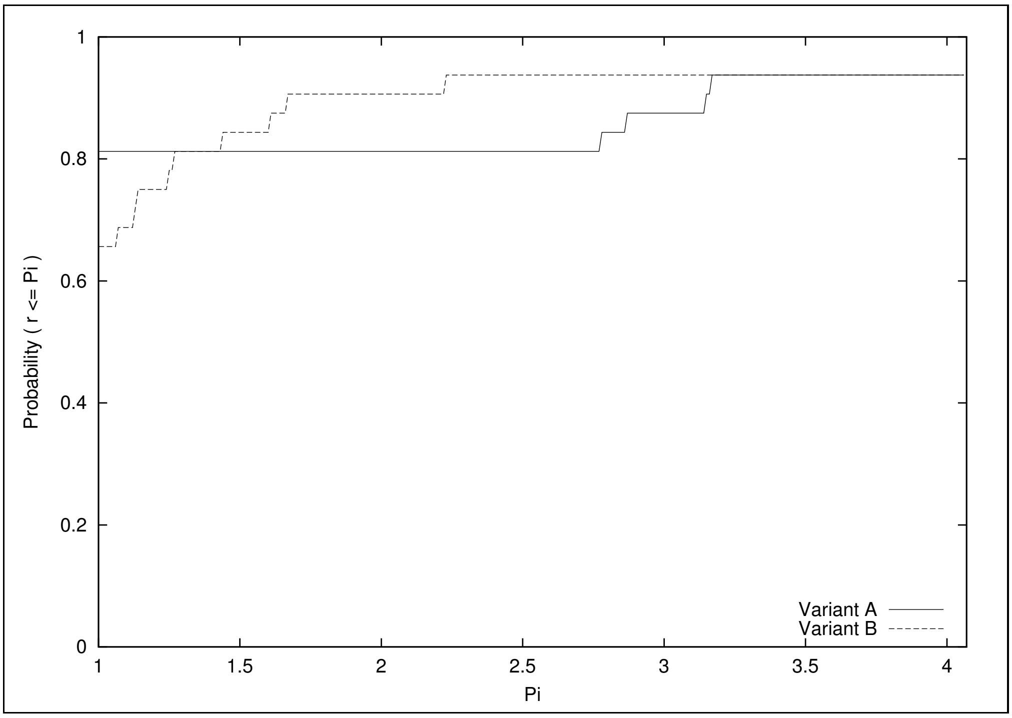


Figure 2.13: Comparison of the number of iterations for Variants A and B on non-singular problems

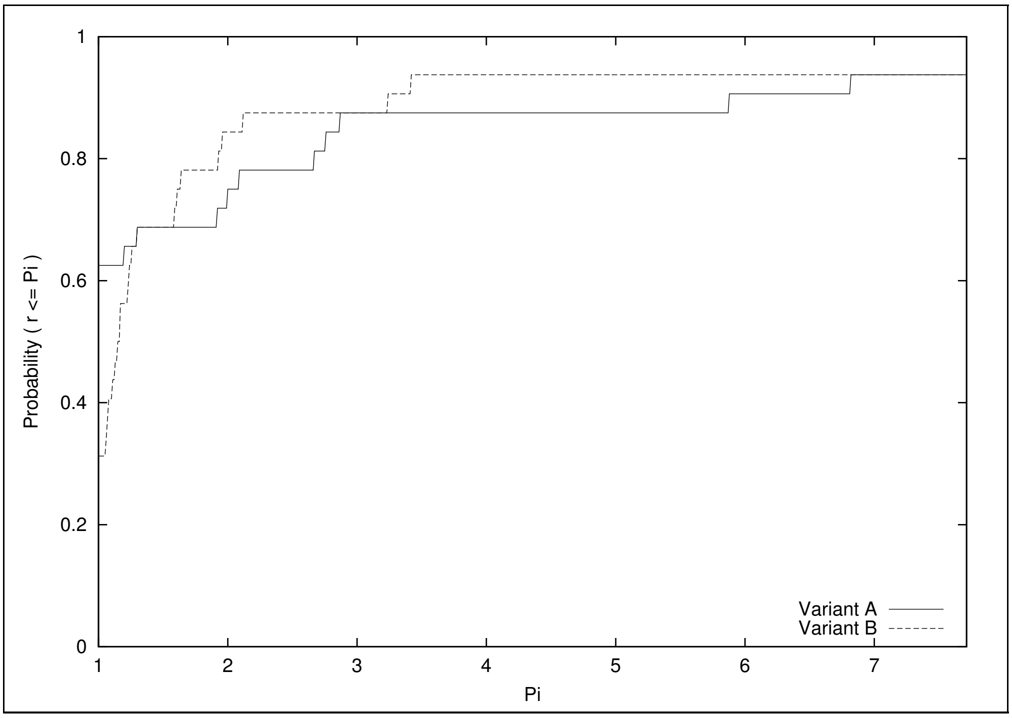


Figure 2.14: Comparison of the CPU time for Variants A and B on non-singular problems

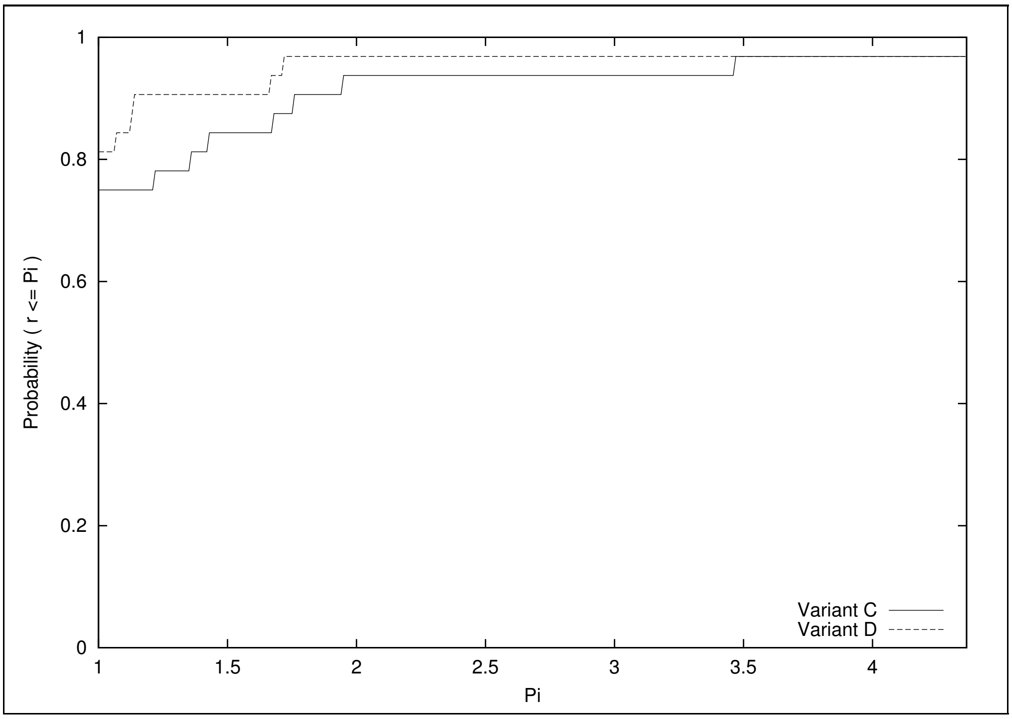


Figure 2.15: Comparison of the number of iterations for Variants C and D on non-singular problems

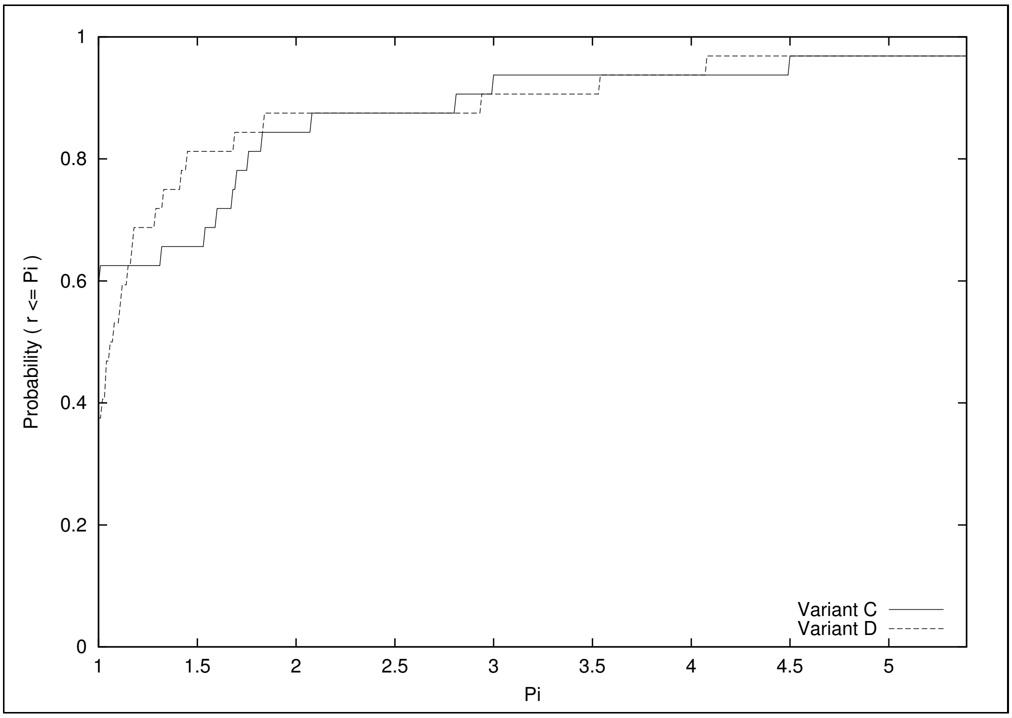


Figure 2.16: Comparison of the CPU time for Variants C and D on non-singular problems

# Chapter 3

# Nonlinear global optimization

Contents

## 3.1 Introduction

We propose a new heuristic for nonlinear global optimization consisting in a variable neighborhood search framework within which we use a modified trust-region algorithm as local search. The proposed method presents the capability to prematurely interrupt the local search if the iterates are converging to a local minimum which has already been visited or if they are reaching an area where no significant improvement can be expected. The neighborhoods as well as the neighbors selection procedure are exploiting the curvature of the objective function. Numerical tests are performed on a set of unconstrained nonlinear problems from the literature. Results illustrate that the new method significantly outperforms existing heuristics from the literature in terms of success rate, CPU time, and number of function evaluations.

The main parts of this chapter are inspired from Bierlaire et al. (2007), which has been submitted to *INFORMS Journal on Computing.*

## 3.2 Motivation and literature review

We are interested in the identification of a global minimum of the nonlinear optimization problem defined by

$$\min_{x \in \mathbb{R}^n} f(x), \tag{3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable. No special structure is assumed on $f$. The vast literature on nonlinear optimization (Bertsekas, 1999, Nocedal and Wright, 1999, Conn et al., 2000, Bierlaire, 2006a to cite a few) focuses on the global convergence of algorithms toward a local optimum, with a fast local convergence. Note that “local” and “global” are used in two different ways in this literature. A point $x^*$ is a global minimum of $f$ if $f(x^*) \leq f(x)$, for all $x \in \mathbb{R}^n$. It is a local minimum of $f$ if there exists $\varepsilon > 0$ such that $f(x^*) \leq f(x)$ for each $x$ such that $\|x - x^*\| \leq \varepsilon$. An algorithm is said to be globally convergent if it converges to a (local) minimum from any starting point. It is locally convergent when it is converging to a (local) minimum when the starting point $x_1$ is in a neighborhood of $x^*$. We refer the reader to the nonlinear optimization literature for more details. In this chapter, we are interested in a heuristic which is (hopefully, not provably) globally convergent toward a global minimum.

Nowadays, there exist efficient and robust methods and softwares to solve unconstrained nonlinear optimization problems (in the sense of identifying a local minimum). In cases when the problems present several local minima, convergence to a global minimum cannot be ensured. There are several practical applications where a global minimum of a nonlinear function is required. Among them, we can cite two important categories:

the maximum likelihood estimation of econometric models (see Chapter 1), and the computation of equilibrium of energy functions in mechanical and chemical engineering.

In econometrics, more and more nonlinear models are being developed to address the complexity of real phenomena under analysis. For instance, discrete choice models are mathematical models used to analyze and predict the behavior of individuals when faced to choice situations. A review of these methods can be found in Chapter 1 (see also Ben-Akiva and Lerman, 1985 and Ben-Akiva and Bierlaire, 2003). The maximum likelihood estimation of those models involves the maximization of nonlinear non-concave functions, exhibiting several local maxima.

The issue of finding a global optimum in the context of econometric models estimation, and in particular discrete choice models, has almost not been addressed in the literature. The standard simulated annealing heuristic is a widely used algorithm in this context (see Goffe et al., 1992, Goffe et al., 1994 ). Dorsey and Mayer (1995) have also proposed genetic algorithms.

Several problems in global optimization arise from chemical and mechanical engineering (see, for instance, Floudas et al., 1999). Lin and Stadtherr (2004) have proposed deterministic approaches for global optimization in the context of parameter estimation of models from computational chemistry and molecular modeling. The identification of a global optimum of continuous functions is also needed in phase equilibrium calculations via Gibbs free energy minimization (Teh and Rangaiah, 2003) as well as in protein structure prediction (Klepeis et al., 2003). In the context of mechanical engineering, global optimization problems occur in the robust design of structures (see, for instance, Sandgren and Cameron, 2002).

The literature on nonlinear global optimization can be divided into two categories: deterministic and exact approaches on the one hand, and heuristics and meta-heuristics on the other hand.

The most important deterministic approaches are (i) methods based on real algebraic geometry (see Lasserre, 2001, Henrion and Lasserre, 2003 and Lasserre, 2004), (ii) exact algorithms as the adaptation of Branch and Bound proposed by Androulakis et al. (1995), and (iii) interval analysis (see Hansen and Walster, 2003 for a review of these methods). Some other approaches exploit the structure of $f$ and the fact that a nonconvex function can be described by the difference of convex functions. The DC (difference of convex functions) programming and related DCA algorithms have been applied successfully to global optimization of nonconvex functions (Horst and Thoai, 1999, and Le and Pham, 2005).

The use of heuristics to address in practice the difficult problem of global nonlinear optimization has been intensive for several decades and is still relevant (see, for instance, Hedar and Fukushima, 2004, Hedar and Fukushima, 2006, and Mladenovic et al., 2006). Many approaches consist in hybridizing derivative-free methods with heuristics originally designed for discrete optimization problems. For example, Hedar and Fukushima (2002) have developed an hybrid simulated annealing method by combining a Nelder-Mead algorithm with simulated annealing.

The same authors have proposed another simulated annealing algorithm using other derivative-free methods like the approximate descent direction (ADD) and pattern search (see Hedar and Fukushima, 2004) as well as a new tabu search method which is "directed" by direct search methods in Hedar and Fukushima (2006). Franzè and Speciale (2001) have adapted a tabu search method using a pattern search algorithm. Tabu search has also been combined with both scatter search and direct search (see Glover, 1994). Recently, Mladenovic et al. (2006) have proposed an adaptation of the variable neighborhood search (VNS) heuristic for unconstrained nonlinear optimization problems using random distributions to compute neighbors. Vaz and Vicente (to appear) have developed a hybrid algorithm that combines a particle swarm heuristic with a pattern search method.

Continuous adaptations of classical heuristics in discrete optimization are also proposed for nonlinear global optimization. The simulated annealing algorithm has been widely adapted for the continuous case (see, for example, Chelouah and Siarry, 1997, Locatelli, 2000). New algorithms based on tabu search (see Chelouah and Siarry, 2000b, Battiti and Tecchiolli, 1996) and genetic algorithms (see Chelouah and Siarry, 2000a, and Chelouah and Siarry, 2003) have also been derived.

Most of the heuristics designed for nonlinear global optimization are inspired from the discrete optimization literature. It is interesting to note that these papers are using either simple local searches (random search for instance) or derivative-free methods such as direct search strategies like Nelder-Mead. These algorithms are not making use of first and second order derivatives. We have only found few algorithms using more advanced local searches. Renders and Flasse (1996) have hybridized genetic algorithms with a quasi-Newton method, while Dekkers and Aarts (1991) made an hybridation of simulated annealing with both steepest descent and quasi-Newton methods as local search. Recently, Mladenovic et al. (2006) have proposed a variable neighborhood search framework for nonlinear global optimization, within which efficient algorithms from nonlinear optimization are used. It is one of the rare approaches where an efficient local algorithm for nonlinear optimization is adapted to global optimization.

The heuristic proposed in this chapter is directly inspired by state-of-the-art algorithms for nonlinear optimization, and state-of-the-art heuristics in discrete optimization.

Among the wide variety of Newton-like methods proposed in the literature of nonlinear optimization for solving (3.1), we are particularly interested in quasi-Newton methods which use only the gradient of the objective function $f$ to be supplied and construct an approximate second order model of $f$ (see Nocedal and Wright, 1999). By measuring changes in the gradient $\nabla f$ at previous iterates, they perform a secant approximation of the Hessian matrix $\nabla^2 f$ and exhibit a fast local convergence rate (typically superlinear). As the second order derivatives are not required, they represent a good tradeoff between fast convergence and low computational burden.

In addition to the use of approximated derivatives, we want to ensure convergence from remote starting points in order to get a practical method for finding local minima of (3.1). Global convergence can be enforced using specific techniques. On the one hand, linesearch strategies control, at each iteration of the optimization algorithm, the step length taken in the Newtonian search direction (see, for example, Nocedal and Wright, 1999). On the other hand, trust-region methods approximately compute the minimum of a quadratic model, centered at the current iterate $x_k$, in an appropriate neighborhood of $x_k$ called the trust-region (see Conn et al., 2000). More recently, filter-trust-region methods have been proposed by Fletcher and Leyffer (2002), as an extension of the trust-region framework.

In this chapter, we adopt a trust-region algorithm using a quasi-Newton framework for constructing the quadratic model of the objective function.

In discrete optimization, local search heuristics operate in a *search space* $S$, also called a *solution space*. The elements of this space are called *solutions*. For every solution $s \in S$, a neighborhood $\mathcal{N}(s) \subset S$ is defined. A local search method starts at an initial solution, and then moves repeatedly from the current solution to a neighbor solution in order to try to find better solutions, measured by an appropriate objective function. The most popular local search methods are: simulated annealing (see Kirkpatrick et al., 1983), tabu search, which was originally proposed by Glover (1986) and Hansen (1986), and variable neighborhood search (see Mladenovic and Hansen, 1997). A more recent local search heuristic is the *variable space search* algorithm proposed by Hertz et al. (2006), which uses not only several neighborhoods, but also several objective functions and several search spaces.

Evolutionary heuristics encompass various algorithms such as genetic algorithms (Davidon, 1991), scatter search (Glover, 1998), ant systems (Dorigo and Blum, 2005) and adaptive memory algorithms (Rochat and Taillard, 1995). They can be defined as iterative procedures that use a central memory where information is collected during the search process. Each iteration is made of two complementary phases which modify the central memory. In the cooperation phase, a *recombination operator* is used to create new offspring solutions, while in the self-adaptation phase, the new offspring solutions are modified individually. The output solutions of the self-adaptation phase are used for updating the content of the central memory. The most successful evolutionary heuristics are hybrid algorithms in the sense that a local search technique (tabu search for example) is used during the self-adaptation phase.

In this chapter, we propose a new heuristic inspired by the Variable Neighborhood Search (VNS) framework originally proposed by Mladenovic and Hansen (1997), which is one of the most recent and efficient heuristics for discrete optimization. The choice of this heuristic is motivated by its ability to widely explore the solution space ($\mathbb{R}^n$ in our case) thanks to the use of several types of neighborhoods. VNS proceeds, for example, by exploring increasingly distant neighborhoods from the current solution. It is also simple to adapt and implement as it only requires two elements: a list of neighborhoods and a local search algorithm. Moreover, a direct and simple adaptation of the VNS algorithm of Mladenovic and Hansen (1997) has recently been proposed for unconstrained continous global optimization by Mladenovic et al. (2006) and has shown very encouraging results compared to other existing approaches and heuristics.

Our method combines a VNS framework with a trust-region algorithm. The philosophy of our approach is to diversify the set of iterates in order to increase the probability of finding a global minimum of (3.1) and to prematurely interrupt the local search if it is converging to a local minimum which has already been visited or if the iterates are reaching an area where no significant improvement can be expected.

The chapter is organized as follows. In Section 3.3, we present our VNS algorithm. Intensive numerical experiments have been conducted, and the results are presented in Section 3.4. We conclude and give some perspectives for future research in Section 3.5.

## 3.3 Algorithm

The local search procedure plays a significant role in our algorithm. In particular, we propose a framework where the local search procedure has the ability to prematurely interrupt its iterations, in order to save computational efforts.

In the following, we refer to the local search procedure as

$$(\text{SUCCESS}, y^*) \leftarrow \text{LS}(y_1, \ell_{\max}, \mathcal{L}), \tag{3.2}$$

where $y_1$ is the starting point of the local search, $\ell_{\max}$ is the maximum number of iterations, $\mathcal{L} = (x_1^*, x_2^*, \ldots)$ is a list of already identified local minima. If this set is non empty, the local search may be prematurely interrupted if it is likely to converge to an already identified local minimum. If the set is empty, the local search converges to a local minimum, except in the presence of numerical problems. SUCCESS is a boolean variable which is true if the procedure has converged to the local minimum $y^*$, and false if the method has failed to converge, or has been prematurely interrupted. If SUCCESS $=$ false, $y^*$ is irrelevant. The local search procedure is globally convergent, so that failure to converge may be due only to severe numerical problems, or to a small value of $\ell_{\text{max}}$. We describe our specific local search procedure in Sections 3.3.1 and 3.3.2.

A VNS heuristic requires also a procedure to define neighbors of a current iterate. We adopt the conventional structure of nested neighborhoods $\mathcal{N}_k(x)$, $k = 1, \dots, n_{\text{max}}$, where $\mathcal{N}_k(x) \subset \mathcal{N}_{k+1}(x) \subseteq \mathbb{R}^n$, for each $k$, and $n_{\text{max}}$ is typically of the order of 5. For each $k$, we use a procedure providing a list of $p$ neighbors of $x$ (typically, $p = 5$) within $\mathcal{N}_k(x)$, that we denote by

$$(z_1, z_2, \dots, z_p) = \text{NEIGHBORS}(x, k). \tag{3.3}$$

We describe two specific neighbor generation procedures in Section 3.3.3.

The VNS framework we are proposing can be described as follows.

**Initialization** The algorithm must be started from $x_1^*$, which is a local mimimum of $f$. We propose two different ways to obtain $x_1^*$: a *cold start* and a *warm start* procedures. A cold start happens when the user provides a local optimum, or when the local search procedure is run once until convergence. In this case, the set of visited local minima $\mathcal{L}$ is initialized as $\mathcal{L} = \{x_1^*\}$. The warm start procedure, which has shown to be useful in practice, proceeds as follows.

1. Initialize the set of local minima as $\mathcal{L} = \emptyset$.
2. Generates randomly $m$ points $y_j$, $j = 1, \dots, m$.
3. Apply $m$ times the local search procedure, that is

$$(\text{SUCCESS}_j, y_j^*) \leftarrow \text{LS}(y_j, \ell_{\text{small}}, \emptyset), \tag{3.4}$$

   where typical values of the parameters used in our experiments are $m = 5$ and $\ell_{\text{small}} = 20$.
4. If SUCCESS$_j$ is true, then a local minimum has been identified, and $\mathcal{L} = \mathcal{L} \cup \{y_j^*\}$.
5. Select $y_1 = \text{argmin}_{j=1,\dots,m} f(y_j^*)$, the best point generated by the above procedure.
6. Apply the local search procedure from $y_1$, that is

$$(\text{SUCCESS}, x_1^*) \leftarrow \text{LS}(y_1, \ell_{\text{large}}, \emptyset), \tag{3.5}$$

where typical values of $\ell_{\text{large}}$ range from 200 to 1000, depending on the size $n$ of the problem. If $y_1$ is a local minimum, (3.5) is not applied. Also, we update

$$\mathcal{L} = \mathcal{L} \cup \{x_1^*\}. \tag{3.6}$$

If this last local search fails (that is, SUCCESS is false), we declare the heuristic has failed, and stop.

The best iterate found is denoted by $x_{\text{best}}^1$ and is initialized by $x_1^*$.

The iteration counter $k$ is initialized to 1.

**Stopping criteria** The iterations are interrupted when one of the following criteria is verified.

1. The last neighborhood was unsuccessfully investigated, that is $k > n_{\text{max}}$.
2. The CPU time exceeds a given threshold $t_{\text{max}}$, typically 30 minutes (1.8K seconds).
3. The number of function evaluations exceeds a given threshold $\text{eval}_{\text{max}}$, typically $10^5$.

**Main loop** For each VNS phase, we apply the following steps.

1. Generate neighbors of $x_{\text{best}}^k$:

$$(z_1, z_2, \ldots, z_p) = \text{NEIGHBORS}(x_{\text{best}}^k, k). \tag{3.7}$$

2. The local search procedure is applied $p$ times, starting from each generated neighbor, that is, for $j = 1, \ldots, p$,

$$(\text{SUCCESS}_j, y_j^*) \leftarrow \text{LS}(z_j, \ell_{\text{large}}, \mathcal{L}). \tag{3.8}$$

3. If all local search procedures have been interrupted, that is if $\text{SUCCESS}_j =$ false, for $j = 1, \ldots, p$, we have two variants:

**Economical** We set $k = k + 1$ and proceed to the next VNS phase.

**Conservative** We apply the local search to convergence from the best point identified by the procedure, that is

$$(\text{SUCCESS}, y^*) \leftarrow \text{LS}(z^*, \ell_{\text{large}}, \emptyset), \tag{3.9}$$

where $z^*$ is such that $f(z^*) \leq f(y_j^*)$, $j = 1, \ldots, p$. If SUCCESS = true, we update the set of local optima: $\mathcal{L} = \mathcal{L} \cup \{y^*\}$.

4. Otherwise, we update the list of local minima, that is for each $j$ such that $\text{SUCCESS}_j = \text{true}$,
$$\mathcal{L} = \mathcal{L} \cup \{y_j^*\}. \tag{3.10}$$

5. We define $x_{\text{best}}^{k+1}$ as the best point in $\mathcal{L}$, that is $x_{\text{best}}^{k+1} \in \mathcal{L}$ and
$$f(x_{\text{best}}^{k+1}) \leq f(x), \text{ for each } x \in \mathcal{L}. \tag{3.11}$$

6. If $x_{\text{best}}^{k+1} = x_{\text{best}}^{k}$, then we could not improve the best solution during this VNS phase. We must investigate the next neighborhood. We set $k = k + 1$ and we proceed to the next VNS phase.

7. Otherwise, we have found a new candidate for the global optimum. The neighborhood structure is reset, that is, $x_{\text{best}}^{1} = x_{\text{best}}^{k+1}$, $k = 1$ and we proceed to the next VNS phase.

**Output** The output is the best solution found during the algorithm, that is $x_{\text{best}}^{k}$.

### 3.3.1 Local search

We now describe our local search procedure (3.2). It is based on a trust region framework (see Conn et al., 2000).

A trust-region algorithm is an iterative numerical procedure in which the objective function $f$ is approximated in a suitable neighborhood of the current iterate (we call it the trust-region). More formally, it can be described as follows.

**Initialization** Initialize the radius of the trust region[1] $\Delta_1$, the iteration counter $\ell = 1$ and $H_1 = I$, the identity matrix.

**Model definition** Define a quadratic model of $f$ around $y_\ell$, that is, for $s \in \mathbb{R}^n$,
$$m_\ell(y_\ell + s) = f(y_\ell) + \nabla f(y_\ell)^T s + \frac{1}{2} s^T H_\ell s. \tag{3.12}$$

**Step computation.** Compute a tentative step $s_\ell$ within the trust region, that sufficiently reduces the model $m_\ell$. This is obtained by solving (not necessarily to optimality) the so-called *trust-region subproblem*:
$$\begin{cases} \min_s m_\ell(y_\ell + s) \\ \text{s.t. } \|s\|_2 \leq \Delta_\ell. \end{cases} \tag{3.13}$$

We use the truncated conjugate gradient algorithm described by Conn et al. (2000, chap. 7) (see also Toint, 1981 and Steihaug, 1983).

[1] In our tests, we have used the procedure proposed by Sartenaer (1997), but any arbitrary, strictly positive, value can be used.

**Acceptance of the tentative step** Compute $f(y_\ell + s_\ell)$ and define

$$\rho_\ell = \frac{f(y_\ell) - f(y_\ell + s_\ell)}{m_\ell(y_\ell) - m_\ell(y_\ell + s_\ell)}. \tag{3.14}$$

If $\rho_\ell \geq 0.1$, the tentative step is accepted, and we define $y_{\ell+1} = x_\ell + s_\ell$; otherwise it is rejected and $y_{\ell+1} = y_\ell$.

**Trust-region radius update.** Set

$$\Delta_{k+1} = \begin{cases} \max(2\|s_\ell\|_2, \Delta_\ell) & \text{if } \rho_\ell \geq 0.9, \\ \Delta_\ell & \text{if } \rho_\ell \in [0.1, 0.9), \\ 0.5\|s_\ell\|_2 & \text{if } \rho_\ell \in [0, 0.1). \end{cases}$$

If it happens that $\rho_\ell < 0$, the adequation between the model and the objective function is so poor that we consider it as a special case. In this case, we use the technique described in Conn et al. (2000, chap. 17).

**Update the hessian approximation** using the symmetric rank one (SR1) formula when the tentative step is accepted:

$$H_\ell = H_{\ell-1} + \frac{(g_{\ell-1} - H_{\ell-1}d_{\ell-1})(g_{\ell-1} - H_{\ell-1}d_{\ell-1})^T}{(g_{\ell-1} - H_{\ell-1}d_{\ell-1})^T d_{\ell-1}} \tag{3.15}$$

where $d_{\ell-1} = y_\ell - y_{\ell-1}$ and $g_{\ell-1} = \nabla f(y_\ell) - \nabla f(y_{\ell-1})$.

Note that $H_\ell$ is not necessarily positive definite when using SR1.

**Stopping criteria** The algorithm is interrupted in any one of the below cases.

- If $\ell \geq \ell_{\max}$, the maximum number of iterations is reached. Set SUCCESS = false, $y^* = y_\ell$ and STOP.
- If $\|\nabla f(y_\ell)\| \leq 10^{-6}$, the local search has converged to a local minimum up to the desired precision, set SUCCESS = true, $y^* = y_\ell$ and STOP.
- If one of the tests described in Section 3.3.2 is verified, then it is preferable to prematurely interrupt the iterations. We set SUCCESS = false, $y^* = y_\ell$ and STOP.

### 3.3.2 Identification of unpromising convergence

A key feature of our approach is to save computational time by prematurely interrupting the local search iterations if they do not look promising. Note that these tests are applied only if the set $\mathcal{L}$ in (3.2) is not empty.

In order to do so, we combine three criteria. First, we check that the algorithm does not get closer and closer to an already identified local minimum. Second, we check that the gradient norm is not too small when the value of the objective function is far from the value at the best iterate in $\mathcal{L}$. Third, we check if a significant reduction in the objective function is achieved.

More formally, we prematurely interrupt the local search iterations if any one of the following conditions is verified when a tentative step is accepted.

- $\exists x \in \mathcal{L}$ such that $\|y_\ell - x\| \leq 1$,
- $\|\nabla f(y_\ell)\| \leq 10^{-3}$ and $f(y_\ell) - f_{\text{best}} \geq 3$, where $f_{\text{best}}$ is the value of the objective function at the best iterate in $\mathcal{L}$,
- $f(y_\ell) > f(y_{\ell-1}) + 0.3 \nabla f(y_{\ell-1})^T s_{\ell-1}$ and $f(y_\ell) - f_{\text{best}} \geq 3$. This Armijo-like condition is supposed to be more demanding than the sufficient reduction condition, based on (3.14).

Note that the threshold values presented above have been empirically selected based on various tests of the algorithm.

### 3.3.3 Generating neighborhoods

We present now the neighbors generating procedure (3.3). The key idea is to analyze the curvature of $f$ at $x$ through an analysis of the eigenstructure of $H$, the approximation of the second derivatives matrix of $f$ at $x$.

Let $v_1, \ldots, v_n$ be the (normalized) eigenvectors of $H$, and $\lambda_1, \ldots, \lambda_n$ the corresponding eigenvalues. We compute them using a standard QR procedure (see, for instance, Golub and Van Loan, 1996).

Neighbors are generated in direction $w_1, \ldots, w_{2n}$, where $w_i = v_i$ if $i \leq n$, and $w_i = -v_i$ otherwise. The size of the neighborhood is defined as a function of $k$ as follows. If $k = 1$, then $d_k = d^{\text{INIT}}$. If $k > 1$, then $d_k = \gamma d_{k-1}$. We have adopted $d^{\text{INIT}} = 1$ and $\gamma = 1.5$ after various numerical tests. The $p$ neighbors generated by this procedure are of the type

$$z_j = x + \alpha d_k w_i \tag{3.16}$$

where $j = 1, \ldots, p$, $\alpha$ is randomly drawn using a uniform distribution between 0.75 and 1, and $i$ is the index of a selected direction.

The indices $i$ for the neighbor generation process are selected according to a sequence of random draws among the $2n$ possible values. We immediately note that the same direction can be selected more than once. In this case, the randomness of $\alpha$ practically guarantees that different neighbors are generated.

The idea is to assign more probability to directions such that the curvature of the function is larger. Indeed, it is hoped that moving in a direction of high curvature increases the chance to jump toward another valley. Taking directions associated with small curvature might cause the iterates to get stuck in a valley where significant improvement cannot be achieved. More formally, the probability for $w_i$ to be selected is given by

$$P(w_i) = P(-w_i) = \frac{e^{\beta \frac{\lambda_i}{d_k}}}{2 \sum_{j=1}^{n} e^{\beta \frac{\lambda_j}{d_k}}}. \tag{3.17}$$

The probability distribution depends on $\beta$ which can be viewed as a weight factor associated with the curvature. A value of $\beta = 0$ corresponds to a uniform distribution, where the curvature is actually ignored. A high value of $\beta$ affects all the mass to the two directions with highest curvature, and zero probability elsewhere. In our tests, we have selected a value of $\beta = 0.05$. Note that the curvature of the function is a local information, which may not be relevant for large neighborhoods. The role of $d_k$ in (3.17) is to decrease the impact of the curvature and to converge toward a uniform distribution as the size of the neighborhoods grows.

## 3.4 Numerical experiments

We have performed intensive numerical tests on a set of problems from the literature (see, for instance, Hedar and Fukushima, 2002, and Chelouah and Siarry, 2003). More precisely, we have used a total of 25 optimization problems corresponding to 15 different test functions described in Appendix A.1. The functions we use to challenge our algorithm exhibit very different and specific shapes. Most of the functions present several local minima. Some of them have many crowded local minima such as Shubert (SH) and Rastrigin (RT) functions. Easom (ES) function has its global minimum lying in a very narrow hole while the well-known Rosenbrock ($R_n$) presents a narrow valley. Smooth and more standard functions like De Joung (DJ) function or Zakharov ($Z_n$) function have also been used.

Note that for each test problem, a search space is provided in which initial points can be randomly selected.

### 3.4.1 Performance analysis

All variants of our algorithm and test functions have been implemented with the package Octave (see `www.octave.org` or Eaton, 1997b) and computations have been done on a desktop equipped with 3GHz CPU, in double precision.

For each test problem, 100 runs have been performed with our VNS algorithm, except for larger size instances ($n \geq 50$) when only 20 trials have been made. A run is considered to be successful if the VNS algorithm finds a global minimum of the problem. The *warm start* procedure described in Section 3.3 has been used for all reported tests of our VNS. The random starting points involved in this procedure have been randomly selected in the search domain associated with the problem (see Appendix A.1).

We consider two measures of performance: the average percentage of success and the average number of function evaluations across successful runs. Performances of competiting algorithms have been obtained from the literature.

For the sake of fair comparison with competitors, the gradient of the objective function used in the local search is computed by using finite differences, requiring $n$ additional function evaluations. Consequently, a single iteration of our local search algorithm requires $n + 1$ function evaluations. In practice, analytical gradients should be used to improve the efficiency of our algorithm.

We are presenting some results with the method proposed by Dolan and Moré (2002) and described in Section 2.6.2. The performance index is the average number of function evaluations across successful runs.

Note that the sum of $\rho_a(1)$ values for all algorithms $a$ considered in a given profile may exceed 1 in the case that some algorithms perform exactly the same on some of the tested problems. Note also that $\rho_a(\pi)$ goes to 1 one as $\pi$ grows. Methods such that $\rho_a(\pi)$ converges fast to 1 are considered more efficient.

### 3.4.2 Variants and competitors

We consider three variants of our algorithm described in Section 3.3:

1. the main method is called *VNS*. It uses the *economical* variant described in Section 3.3 and consequently never applies the local search without potentially interrupting it prematurely. Also, the probabilistic formula (3.17) is used with $\beta = 0.05$.

2. the conservative method is called $\text{VNS}_a$. It uses the *conservative* variant described in Section 3.3 in which the local search is applied without premature stop if all local searches have been stopped.

3. the third variant is called $\text{VNS}_b$. It sets $\beta = 0$ in (3.17), that is, it uses equal probabilities for the neighbors generation procedure.

We compare our method with the following methods from the literature:

1. Direct Search Simulated Annealing (DSSA), see Hedar and Fukushima (2002).

2. Continuous Hybrid Algorithm (CHA), see Chelouah and Siarry (2003).
3. Simulated Annealing Heuristic Pattern Search (SAHPS), see Hedar and Fukushima (2004).
4. Directed Tabu Search (DTS), see Hedar and Fukushima (2006).
5. General Variable Neighborhood Search (GVNS), see Mladenovic et al. (2006). Note that the authors report the number of function evaluations necessary to find the global minimum (the first one of them if several have been identified during the optimization process) and not the number of function evaluations performed before the algorithm stops. The measure of performance is thus slightly different.

The respective features of these algorithms have been described in Section 3.2. Note that we have chosen the DTS algorithm with Adaptative Pattern Search (APS), called $\mathrm{DTS}_{\mathrm{APS}}$, which was the best of two variants proposed by Hedar and Fukushima (2006).

### 3.4.3 Tests

In the tables presented in this section, some of the cells, corresponding to competitors, are empty when the information was not reported in the paper.

Table 3.1 gives the number of successes over the 100 runs for 25 problems. Note that we do not report results for variants of our algorithm, $\mathrm{VNS}_a$ and $\mathrm{VNS}_b$, as the results are very similar.

Table 3.2 gives the average number of function evaluations for successful runs on the same 25 problems. In this table, results for all 3 variants of our VNS are included. The comparison between VNS and GVNS on their 10 common problems is available in a specific table (see Table 3.4) as the measure of performance is slightly different. While Tables 3.2 and 3.4 present absolute values for the average number of function evaluations, Tables 3.3 and 3.5 give the corresponding normalized values with respect to our *VNS* algorithm.

Finally, Table 3.6 provides a few results available in terms of CPU time for DTS heuristic on large size problems. CPU time comparison is always complicated. As DTS has been published in 2006, we believe that it illustrates well the good performance of our algorithm.

#### 3.4.3.1 Comparison of VNS with competitors except GVNS

We first focus on the five first columns of Tables 3.1 and 3.2 for problems of small size (skipping the three last rows).

| Problem | VNS | CHA | DSSA | DTS | SAHPS | GVNS |
|---|---|---|---|---|---|---|
| RC | 100 | 100 | 100 | 100 | 100 | 100 |
| ES | 100 | 100 | 93 | 82 | 96 | |
| RT | 84 | 100 | 100 | | 100 | |
| SH | 78 | 100 | 94 | 92 | 86 | 100 |
| $R_2$ | 100 | 100 | 100 | 100 | 100 | 100 |
| $Z_2$ | 100 | 100 | 100 | 100 | 100 | |
| DJ | 100 | 100 | 100 | 100 | 100 | |
| $H_{3,4}$ | 100 | 100 | 100 | 100 | 95 | 100 |
| $S_{4,5}$ | 100 | 85 | 81 | 75 | 48 | 100 |
| $S_{4,7}$ | 100 | 85 | 84 | 65 | 57 | |
| $S_{4,10}$ | 100 | 85 | 77 | 52 | 48 | 100 |
| $R_5$ | 100 | 100 | 100 | 85 | 91 | |
| $Z_5$ | 100 | 100 | 100 | 100 | 100 | |
| $H_{6,4}$ | 100 | 100 | 92 | 83 | 72 | 100 |
| $R_{10}$ | 100 | 83 | 100 | 85 | 87 | 100 |
| $Z_{10}$ | 100 | 100 | 100 | 100 | 100 | |
| HM | 100 | | 100 | | | |
| $GR_6$ | 100 | | 90 | | | |
| $GR_{10}$ | 100 | | | | | 100 |
| CV | 100 | | 100 | | | |
| DX | 100 | | 100 | | | |
| MG | 100 | | | | | 100 |
| $R_{50}$ | 100 | 79 | | 100 | | |
| $Z_{50}$ | 100 | 100 | | 0 | | |
| $R_{100}$ | 100 | 72 | | 0 | | |

Table 3.1: Percentage of success

| Problem | VNS | CHA | DSSA | DTS | SAHPS | $VNS_a$ | $VNS_b$ |
|---|---|---|---|---|---|---|---|
| RC | 153 | 295 | 118 | 212 | 318 | 179 | 165 |
| ES | 167 | 952 | 1442 | 223 | 432 | 249 | 237 |
| RT | 246 | 132 | 252 |  | 346 | 340 | 234 |
| SH | 366 | 345 | 457 | 274 | 450 | 630 | 424 |
| DJ | 104 | 371 | 273 | 446 | 398 | 104 | 104 |
| $H_{3,4}$ | 249 | 492 | 572 | 438 | 517 | 292 | 268 |
| $H_{6,4}$ | 735 | 930 | 1737 | 1787 | 997 | 1036 | 759 |
| $S_{4,5}$ | 583 | 698 | 993 | 819 | 1073 | 769 | 589 |
| $S_{4,7}$ | 596 | 620 | 932 | 812 | 1059 | 752 | 591 |
| $S_{4,10}$ | 590 | 635 | 992 | 828 | 1035 | 898 | 664 |
| $R_2$ | 556 | 459 | 306 | 254 | 357 | 847 | 618 |
| $Z_2$ | 251 | 215 | 186 | 201 | 276 | 273 | 280 |
| $R_5$ | 1120 | 3290 | 2685 | 1684 | 1104 | 2197 | 1157 |
| $Z_5$ | 837 | 950 | 914 | 1003 | 716 | 866 | 831 |
| $R_{10}$ | 2363 | 14563 | 16785 | 9037 | 4603 | 4503 | 2358 |
| $Z_{10}$ | 1705 | 4291 | 12501 | 4032 | 2284 | 1842 | 1754 |
| HM | 335 |  | 225 |  |  | 388 | 359 |
| $GR_6$ | 807 |  | 1830 |  |  | 1011 | 831 |
| CV | 854 |  | 1592 |  |  | 1346 | 782 |
| DX | 2148 |  | 6941 |  |  | 3057 | 2243 |
| $R_{50}$ | 11934 | 55356 |  | 510505 |  |  |  |
| $Z_{50}$ | 17932 | 75520 |  | $177125^*$ |  |  |  |
| $R_{100}$ | 30165 | 124302 |  | 3202879 |  |  |  |

Superscript $^*$ means that DTS only obtains points close to the global minimum

Table 3.2: Average number of function evaluations

| Problem | VNS | CHA | DSSA | DTS | SAHPS | $VNS_a$ | $VNS_b$ |
|---|---|---|---|---|---|---|---|
| RC | 1 | 1.93 | 0.77 | 1.39 | 2.08 | 1.17 | 1.08 |
| ES | 1 | 5.70 | 8.63 | 1.34 | 2.59 | 1.49 | 1.42 |
| RT | 1 | 0.54 | 1.02 |  | 1.41 | 1.38 | 0.95 |
| SH | 1 | 0.94 | 1.25 | 0.75 | 1.23 | 1.72 | 1.16 |
| DJ | 1 | 3.57 | 2.62 | 4.29 | 3.83 | 1 | 1 |
| $H_{3,4}$ | 1 | 1.98 | 2.30 | 1.76 | 2.08 | 1.17 | 1.08 |
| $H_{6,4}$ | 1 | 1.27 | 2.36 | 2.43 | 1.36 | 1.41 | 1.03 |
| $S_{4,5}$ | 1 | 1.20 | 1.70 | 1.40 | 1.84 | 1.32 | 1.01 |
| $S_{4,7}$ | 1 | 1.04 | 1.56 | 1.36 | 1.78 | 1.26 | 0.99 |
| $S_{4,10}$ | 1 | 1.08 | 1.68 | 1.4 | 1.75 | 1.52 | 1.13 |
| $R_2$ | 1 | 0.83 | 0.55 | 0.46 | 0.64 | 1.52 | 1.11 |
| $Z_2$ | 1 | 0.86 | 0.74 | 0.80 | 1.10 | 1.09 | 1.12 |
| $R_5$ | 1 | 2.94 | 2.40 | 1.50 | 0.99 | 1.96 | 1.03 |
| $Z_5$ | 1 | 1.14 | 1.09 | 1.20 | 0.86 | 1.03 | 0.99 |
| $R_{10}$ | 1 | 6.16 | 7.10 | 3.82 | 1.95 | 1.91 | 0.99 |
| $Z_{10}$ | 1 | 2.52 | 7.33 | 2.36 | 1.34 | 1.08 | 1.03 |
| HM | 1 |  | 0.67 |  |  | 1.16 | 1.07 |
| $GR_6$ | 1 |  | 2.27 |  |  | 1.25 | 1.03 |
| CV | 1 |  | 1.86 |  |  | 1.58 | 0.92 |
| DX | 1 |  | 3.23 |  |  | 1.42 | 1.04 |
| $R_{50}$ | 1 | 4.64 |  | 42.78 |  |  |  |
| $Z_{50}$ | 1 | 4.21 |  | $9.88^*$ |  |  |  |
| $R_{100}$ | 1 | 4.12 |  | 106.18 |  |  |  |

Superscript $^*$ means that DTS only obtains points close to the global minimum

Table 3.3: Normalization of average number of function evaluations

| Problem | VNS | GVNS |
|---|---|---|
| RC | 99 | 45 |
| SH | 305 | 623 |
| $R_2$ | 176 | 274 |
| $R_{10}$ | 1822 | 39062 |
| $GR_{10}$ | 1320 | 1304 |
| $H_{3,4}$ | 174 | 385 |
| $H_{6,4}$ | 532 | 423 |
| $S_{4,5}$ | 468 | 652 |
| $S_{4,10}$ | 481 | 676 |
| MG | 17 | 73 |

Table 3.4: Average number of function evaluations - VNS against GVNS

| Problem | VNS | GVNS |
|---|---|---|
| RC | 1 | 0.45 |
| SH | 1 | 2.04 |
| $R_2$ | 1 | 1.56 |
| $R_{10}$ | 1 | 21.44 |
| $GR_{10}$ | 1 | 0.99 |
| $H_{3,4}$ | 1 | 2.21 |
| $H_{6,4}$ | 1 | 0.80 |
| $S_{4,5}$ | 1 | 1.39 |
| $S_{4,10}$ | 1 | 1.41 |
| MG | 1 | 4.29 |

Table 3.5: Normalization of average number of function evaluations - VNS against GVNS

| Problem | VNS | DTS |
|---|---|---|
| $R_{50}$ | 208 | 1080 |
| $Z_{50}$ | 228 | 1043 |
| $R_{100}$ | 1171 | 15270 |

Table 3.6: Average CPU time in seconds - Large size problems

From Table 3.1, we can see that VNS is the most robust algorithm as it achieves a maximal success rate of 100% on almost all problems, actually 18 out of 20. The best challenger of VNS with regard to robustness is CHA as it is able to solve on each run 12 problems among 16. For instance, VNS in the only algorithm able to reach 100% of success on the Shekel $S_{4,n}$ functions.

Table 3.2 shows that VNS presents the lowest average number of function evaluations on the majority of the tested problems. One can also see that the efficiency of VNS on Rosenbrock ($R_n$) and Zakharov ($Z_n$) functions is becoming better and better when the dimension $n$ of the problem increases from 2 to 10. In particular, VNS is able to significantly decrease (up to a factor 7 compared to some methods) the average number of evaluations of $f$ on problems $R_{10}$ and $Z_{10}$.

We now present the performance profiles of all 5 heuristics on 15 common problems (out of the 20 problems in Table 3.2) in Figure 3.1. A zoom for $\pi$ between 1 and 5 is provided in Figure 3.2. The measure of performance is the average number of function evaluations (from Table 3.2). From Figure 3.2, we see that VNS is the best algorithm on 60% of the problems. Moreover, when VNS is not the best algorithm, it remains within a factor around 1.5 of the best method on 90% of the problems. Results are very satisfactory as VNS is the most robust but also the most efficient method among the 5 tested methods.

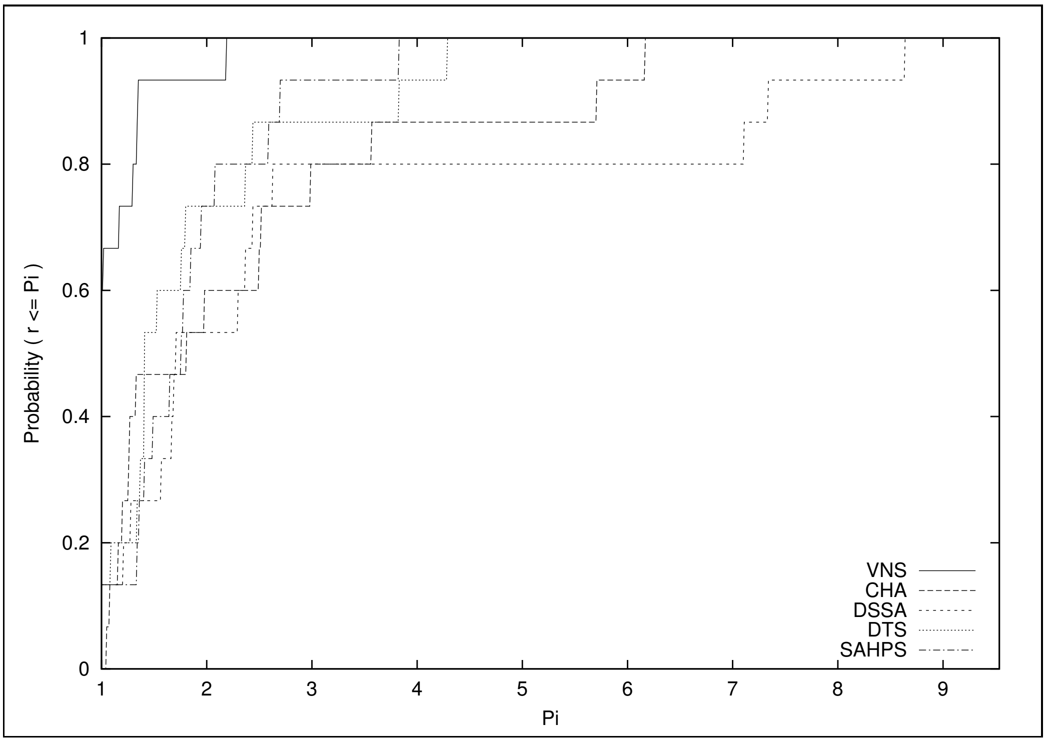


Figure 3.1: Average number of function evaluations

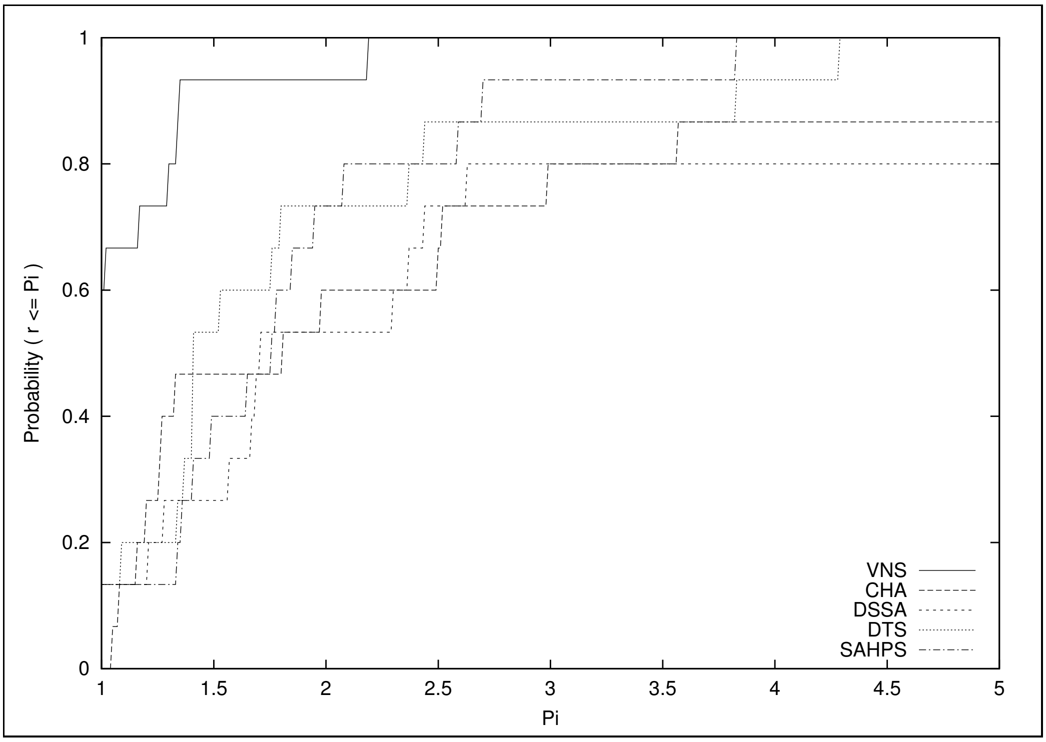


Figure 3.2: Zoom on the average number of function evaluations

As we have already seen, our VNS behaves better and better on Rosenbrock and Zakharov problems as their dimension increases. This motivated us to perform several tests in larger dimension to see if our algorithm can achieve a significant gain in terms of number of function evaluations as well as CPU time. Table 3.1 shows the robustness of VNS compared to CHA and DTS methods on these 3 large size problems while Table 3.2 provides the average number of function evaluations. The performance with regard to the CPU time for VNS and DTS can be found in Table 3.6. VNS is the most robust as well as the most efficient on these 3 problems. Even if CHA is the best competitor, it requires up to 5 times more function evaluations.

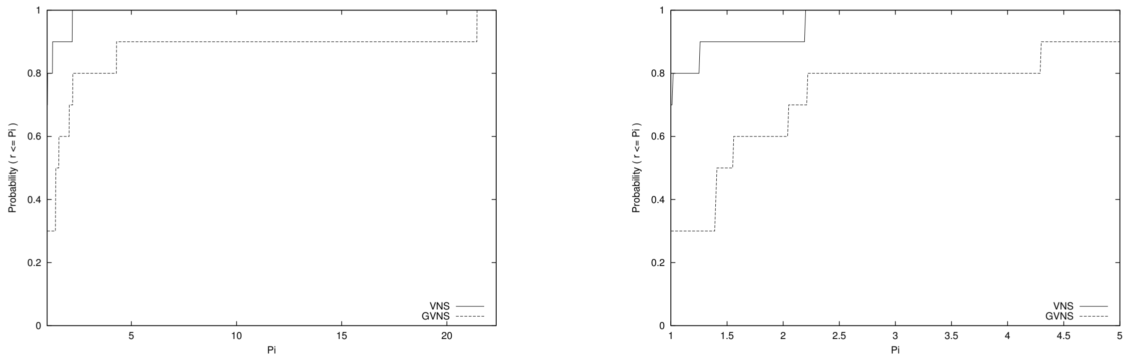
These results are very encouraging for the future use of our algorithm in real applications. Comparing VNS with DTS, we clearly see that there is a computational overhead associated with our algorithm as the gain in CPU time (see Table 3.6) is less impressive compared to the gain in number of function evaluations (see Table 3.2). This is mainly due to our more costly local search and the QR-analysis discussed in Section 3.3.3. However, even if the tested functions are not cumbersome to compute, our method requires significantly less time to identify the global minimum, showing that the additional computational cost of our algorithm is compensated by its better efficiency. Given that, we are confident that the proposed algorithm will reduce the CPU time for solving problems such that the evaluation of the objective function dominates all other computations of numerical algebra.

### 3.4.3.2 Comparison of VNS with GVNS

Now we consider the columns associated with VNS and GVNS in Table 3.1 as well as in Table 3.4. 10 common problems allow to challenge our VNS against the GVNS recently proposed by Mladenovic et al. (2006).

VNS and GVNS exhibit the same high level of robustness on the tested problems. GVNS always reaches the maximal rate of success while VNS attains 100% of success on all common problems, except one. Still, 78 runs on the Shubert (SH) function were successful.

From Table 3.4, we can note that VNS is the most efficient method on 7 out of the 10 problems. It is able to significantly decrease the number of function evaluations required to identify the global minimum of tested problems. When VNS is beaten, it remains within a reasonable factor of GVNS, requiring at worst the double of function evaluations on the Branin RCOS (RC) function. Contrarily to VNS, GVNS can be much slower on several problems. The performance profiles corresponding to Table 3.4 are provided in Figures 3.3(a) and 3.3(b).



(a) Full profile

(b) Zoom

Figure 3.3: Average number of function evaluations - VNS against GVNS

### 3.4.3.3 Comparison of the variants of VNS

We consider the three columns of Table 3.2 associated with the three variants of our VNS.

Comparing VNS with $\mathrm{VNS}_a$, we clearly see that applying a full local search for each

VNS iteration significantly increases the number of evaluations of f, up to a factor 2. As both algorithms are similar in terms of robustness (percentage of runs leading to the global minimum), it means that the tests proposed in Section 3.3.2 allow to reduce the number of function evaluations without deteriorating the capability of finding the global minimum. Moreover, we can argue that applying a classical VNS framework in which full local searches are applied to all neighbors generated within the VNS would definitely be too cumbersome from a computional point of view.

Comparison bewteen VNS and $\mathrm{VNS}_b$ shows that VNS is the best method for most of the 20 tested problems, with an important gain on some problems. From the related performance profile provided in Figure 3.4, it appears that VNS is the fastest method on about 75% of the problems. The gain obtained can be up to a factor of 1.5, meaning that using a purely random selection for search directions in order to compute neighbors may need 50% additional function evaluations compared to the strategy proposed in Section 3.3.3. This strategy prevents the algorithm from getting stucked in a given valley, where no significant improvement can be achieved, and gives the possibility to jump over valleys by using information on the curvature of f.

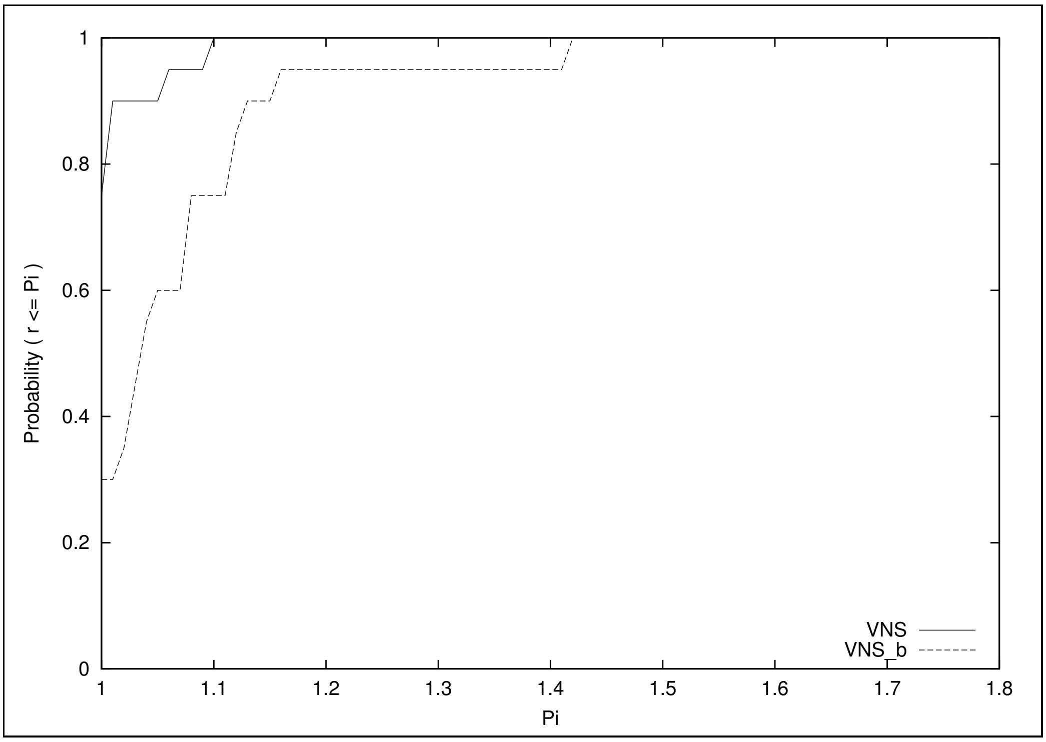


Figure 3.4: Average number of function evaluations for VNS and its variant

## 3.5 Conclusions and perspectives

The chapter deals with nonlinear global optimization. We have proposed a new heuristic dedicated to identify the global minimum of an unconstrained nonlinear problem. Limiting the number of function evaluations is of major importance when the objective function is cumbersome to evaluate.

Within our VNS heuristic, we use an efficient nonlinear optimization algorithm using first and second order derivatives in order to quickly identify a local minimum of the problem. The ability to prematurely stop the local search allows to reduce the number of function evaluations (as well as the CPU time). Information about the function and its derivatives is also used to compute the list of neighborhoods involved in the VNS and to select the associated neighbors. The better use of available and relevant information on $f$ is decisive for the good behavior of the proposed method.

Numerical results obtained with our method are very satisfactory as VNS is the most robust but also the most efficient method on the problems we used in the experiments. The VNS framework makes the algorithm robust by its capabilities of exploration and diversification. The proposed algorithm significantly reduces the number of function evaluations compared to other efficient published methods. This makes the method particularly appealing for problems where the CPU time spent in function evaluations is dominant, such as those involving simulation. The results are consistent with the way the heuristic has been designed.

As a conclusion, we could say that this chapter represents a nice and profitable collaboration and interaction between nonlinear optimization and discrete optimization.

Several improvements should be investigated. We believe that we could have a better estimation of convergence basins of already encountered minima by stocking also other previous iterates and not only local minima. Also, defining $p$ as dynamic from iteration to iteration of the VNS might also be interesting to investigate (see Neveu et al., 2004).

From a numerical point of view, we could implement a more efficient eigen-structure analysis in the VNS algorithm, to reduce the computational cost of the overall method. Other stopping criteria could also be used to compare heuristics challenged in the chapter. For instance, we could see how each method behaves with a given budget of CPU time or a given budget of function evaluations.

Another track of development would be to incorporate the VNS presented in this chapter into an Adaptive Memory Method (AMM) framework (see Rochat and Taillard, 1995) in order to improve the diversification inside our algorithm.

Finally, we could investigate how the ideas presented in this chapter could be tailored to constrained nonlinear global optimization.

# Chapter 4

# Generalized Secant Method

## Contents

## 4.1 Introduction

We propose an extension of secant methods for systems nonlinear equations using a population of previous iterates. Contrarily to classical secant methods, where exact interpolation is used, we prefer a least squares approach to calibrate the linear model. We propose an explicit control of the numerical stability of the method. We show that our approach can lead to an update formula. Then, we propose two globalization techniques in order to obtain a robust quasi-Newton method. Finally, computational comparisons with classical quasi-Newton methods highlight a significant improvement in terms of robustness and number of function evaluations. Globalization strategies are shown to highly improve the robustness of considered secant methods on a set of difficult problems. Damped versions of our approach are showed to be competitive with derivatives-based Newton-Krylov methods.

This chapter is mainly constituted from ideas presented in Bierlaire et al. (to appear) which has been accepted for publication in *European Journal of Operational Research*. The ideas of the GSM algorithm has been originally proposed by Bierlaire and Crittin (2003). The linesearch-filter approach presented in Section 4.4.2 and the associated results shown in Section 4.5 are unpublished.

## 4.2 Problem formulation

We consider the standard problem of identifying the solution of a system of nonlinear equations

$$F(x) = 0 \tag{4.1}$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a differentiable function. Since Newton, this problem has received a tremendous amount of attention. Newton's method and its many variations are still intensively analyzed and used in practice. The idea of Newton-like methods is to replace the nonlinear function $F$ by a linear model, which approximates $F$ in the neighborhood of the current iterate. The original Newton's method invokes Taylor's theorem and uses the gradient matrix (the transpose of which is called the Jacobian) to construct the linear model. When the Jacobian is too expensive to evaluate, secant methods build the linear model based on the secant equation. Because secant methods exhibit a q-superlinear rate of convergence, they have been intensively analyzed in the literature.

The secant equation imposes that the linear model exactly matches the nonlinear function $F$ at two successive iterates. If the number of unknowns $n$ is strictly greater than 1, an infinite number of linear models verify the secant equation. Therefore, each secant method derives a specific update formula which arbitrarily picks one linear model among them. The most common strategies are called "least-change updates" and select the linear model which minimizes the difference between two successive models.

In this chapter, we provide a class of algorithms generalizing these ideas. Instead of using only two successive iterates to determine this linear model, we maintain a "population" of previous iterates. This approach allows all the available information collected through the iterations to be explicitly used for calibrating the model.

An important feature of our method is that we do not impose an exact match between the model and the function. Instead, we use a least squares approach to request that the model fits the function "as well as possible".

In this chapter, we present the class of algorithms based on our method (Section 4.3.2) and propose two globalization strategies in Section 4.4, namely one linesearch and one linesearch-filter technique. This class of algorithms exhibits faster convergence and greater robustness than quasi-Newton methods for most numerical tests that we have performed (Section 4.5) at a cost of substantial linear algebra computation. Therefore it is valuable when the cost of evaluating $F$ is high in comparison with the numerical algebra overhead. We give some conclusions in Section 4.6 and some perspectives in Section 4.7.

## 4.3 Quasi-Newton methods

Quasi-Newton methods consider at each iteration the linear model

$$L_k(x; B_k) = F(x_k) + B_k(x - x_k) \tag{4.2}$$

which approximates $F(x)$ in the neighborhood of $x_k$ and computes $x_{k+1}$ as a solution of the linear system $L_k(x; B_k) = 0$. Consistently with most of the publications on this topic, quasi-Newton methods can be summarized as methods based on the following iterations:

$$x_{k+1} = x_k - B_k^{-1} F(x_k), \tag{4.3}$$

followed by the computation of $B_{k+1}$. The pure Newton's method is obtained with $B_k = J(x_k) = \nabla F(x_k)^T$, the Jacobian of $F$ evaluated at $x_k$, that is a $n \times n$ matrix such that entry $(i, j)$ is $\partial F_i / \partial x_j$. We refer the reader to Dennis and Schnabel (1996) for an extensive analysis of Newton and quasi-Newton methods.

### 4.3.1 Secant methods

Broyden (1965) proposes a quasi-Newton method based on the *secant equations*, imposing the linear model $L_{k+1}$ to exactly match the nonlinear function at iterates $x_k$ and $x_{k+1}$, that is

$$\begin{aligned} L_{k+1}(x_k; B_{k+1}) &= F(x_k), \\ L_{k+1}(x_{k+1}, B_{k+1}) &= F(x_{k+1}). \end{aligned} \tag{4.4}$$

Subtracting these two equations and defining $y_k = F(x_{k+1}) - F(x_k)$ and $s_k = x_{k+1} - x_k$ we obtain the classical secant equation:

$$B_{k+1}s_k = y_k. \tag{4.5}$$

Clearly, if the dimension $n$ is strictly greater than 1, there is an infinite number of matrices $B_{k+1}$ satisfying (4.5). An arbitrary decision must consequently be made. The "least-change secant update" strategy, proposed by Broyden (1965), consists in selecting among the matrices verifying (4.5) the one minimizing variations (in Frobenius norm) between two successive matrices $B_k$ and $B_{k+1}$. It leads to the following update formula

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)\, s_k^T}{s_k^T s_k}. \tag{4.6}$$

As an aside, we note that different techniques can be considered to modify and extend these methods to handle large-scale problems. Namely, limited-memory versions can be considered, where the matrix $B_k$ is never explicitly generated (see Gomes-Ruggiero et al., 1992, Byrd et al., 1994). Other approaches define quasi-Newton methods that preserve the sparsity of the true Jacobian or that can exploit the structural properties of the problem (see, for instance, Diniz-Ehrhardt et al., 2003).

Secant methods have been very successful and widely adopted in the field. However, we believe that the idea of interpolating the linear model at only two iterates and ignoring previous iterates could be too restrictive. Therefore, we propose to use more than two iterates to build the linear model.

This idea has already been considered. Dennis and Schnabel (1996) say that "Perhaps the most obvious strategy is to require the model to interpolate $F(x)$ at other past points... One problem is that the directions tend to be linearly dependent, making the computation of (the approximation matrix) a poorly posed numerical problem". Later, they write "In fact, multivariable generalizations of the secant method have been proposed ... but none of them seem robust enough for general use."

There are few attempts to generalize this approach in the literature. A first generalization of the secant method is the *sequential secant method* proposed by Wolfe (1959) and discussed by Ortega and Rheinboldt (1970). The idea is to impose exact interpolation of the linear model on $n+1$ iterates instead of 2:

$$L_{k+1}(x_{k+1-j}; B_{k+1}) = F(x_{k+1-j}), \quad j = 0, 1, \ldots, n. \tag{4.7}$$

or, equivalently,

$$B_{k+1}s_{k-j} = y_{k-j}, \quad j = 0, 1, \ldots, n-1, \tag{4.8}$$

where $s_i = x_{k+1} - x_i$, and $y_i = F(x_{k+1}) - F(x_i)$, for all $i$. If the vectors $s_k, s_{k-1}, \ldots, s_{k-n+1}$ are linearly independent, there exists exactly one matrix $B_{k+1}$ satisfying (4.8), which is

$$B_{k+1} = Y_{k+1}S_{k+1}^{-1} \tag{4.9}$$

where $Y_{k+1} = (y_k, y_{k-1}, \ldots, y_{k-n+1})$ and $S_{k+1} = (s_k, s_{k-1}, \ldots, s_{k-n+1})$.

Quoting Ortega and Rheinboldt (1970) "...(sequantial methods) are prone to unstable behavior and ... no satisfactory convergence results can be given". Nevertheless Gragg and Stewart (1976) propose a method which avoids instabilities by working with orthogonal factorizations of the involved matrices. Martínez (1979) gives three implementations of the idea proposed by Gragg and Stewart (1976) and some numerical experiments.

Multi-step quasi-Newton methods have been proposed by Moghrabi (1993), Ford and Moghrabi (1997) and Ford (1999) in the context of nonlinear programming. An interpolating path is built based on previous iterates, and used to produce an alternative secant equation. Interestingly, the best numerical results were obtained with no more than two steps.

We believe that the comments about the poor numerical stability of those methods found in major reference texts such as Dennis and Schnabel (1996) and Ortega and Rheinboldt (1970) have not encouraged researchers to pursue these investigatations. We provide here a successful multi-iterate appoach with robust convergence properties and exhibiting an excellent behavior on numerical examples. The idea of using a least squares approach is similar to an idea proposed in the physics litterature by Vanderbilt and Louie (1984), which has inspired other authors in the same field (Johnson, 1988, Eyert, 1996). Bierlaire and Crittin (2006) have used a similar approach for solving noisy large scale transportation problems.

### 4.3.2 Population-based approach

We propose a class of methods calibrating a linear model based on several previous iterates. The difference with existing approaches is that we do not impose the linear model to interpolate the function. Instead, we prefer to identify the linear model which is as close as possible to the nonlinear function, in the least-squares sense.

At each iteration, we maintain a finite population of previous iterates. Without loss of generality, we present the method assuming that all previous iterates $x_0, \ldots, x_{k+1}$ are considered. Our method belongs also to the quasi-Newton framework defined by (4.3), where $B_{k+1}$ is computed as follows.

$$B_{k+1} = \operatorname*{argmin}_{J} \left( \sum_{i=0}^{k} \left\| \omega_{k+1}^{i} F(x_i) - \omega_{k+1}^{i} L_{k+1}(x_i; J) \right\|_2^2 + \left\| J\Gamma - B_{k+1}^{0}\Gamma \right\|_F^2 \right) \tag{4.10}$$

where $L_{k+1}$ is defined by (4.2) and $B_{k+1}^0 \in \mathbb{R}^{n \times n}$ is an a priori approximation of $B_{k+1}$. The role of the second term is to overcome the under-determination of the least squares problem based on the first term and also control the numerical stability of the method. The matrix $\Gamma$ contains weights associated with the arbitrary term $B_{k+1}^0$, and the weights $\omega_{k+1}^i \in \mathbb{R}^+$ are associated with the previous iterates. Equation (4.10) can be written in matrix form as follows:

$$B_{k+1} = \operatorname*{argmin}_{J} \left\| J \begin{pmatrix} S_{k+1} & I_{n\times n} \end{pmatrix} \begin{pmatrix} \Omega & 0_{k\times n} \\ 0_{n\times k} & \Gamma \end{pmatrix} - \begin{pmatrix} Y_{k+1} & B_{k+1}^0 \end{pmatrix} \begin{pmatrix} \Omega & 0 \\ 0 & \Gamma \end{pmatrix} \right\|_F^2$$

where $\Omega \in \mathbb{R}^{k+1}$ is a diagonal matrix with weights $\omega_{k+1}^i$ on the diagonal for $i = 0, \cdots, k$. The normal equations of this least squares problem lead to the following formula:

$$B_{k+1} = B_{k+1}^0 + \left( Y_{k+1} - B_{k+1}^0 S_{k+1} \right) \Omega^2 S_{k+1}^T \left( \Gamma\Gamma^T + S_{k+1}\Omega^2 S_{k+1}^T \right)^{-1}, \tag{4.11}$$

where $Y_{k+1} = (y_k, y_{k-1}, \ldots, y_0)$ and $S_{k+1} = (s_k, s_{k-1}, \ldots, s_0)$.

The role of the a priori matrix $B_{k+1}^0$ is to overcome the possible under-determination of problem (4.10). For example, choosing $B_{k+1}^0 = B_k$ (similarly to classical Broyden-like methods) exhibits good properties. In that case, (4.11) becomes an update formula, and local convergence can be proved (see Bierlaire et al., to appear).

The weights $\omega_{k+1}^i$ capture the relative importance of each iterate in the population. Roughly speaking, they should be designed in the lines of the assumptions of Taylor's theorem, that is assigning more weight to points close to $x_{k+1}$, and less weight to points which are far away. The matrix $\Gamma$ captures the importance of the arbitrary terms defined by $B_{k+1}^0$ for the identification of the linear model. The weights have to be finite, and $\Gamma$ must be such that

$$\Gamma\Gamma^T + S_{k+1}\Omega^2 S_{k+1}^T \tag{4.12}$$

is safely positive definite. To ensure this property we describe below three possible approaches for choosing $\Gamma\Gamma^T$: the *geometrical approach*, based on specific geometric properties of the population, the *subspace decomposition* approach, decomposing $\mathbb{R}^n$ into the subspace spanned by the columns of $S_{k+1}$ and its orthogonal complement, and the *numerical approach*, designed to guarantee a numerically safe positive definiteness of (4.12).

The *geometrical approach* assumes that $n + 1$ members of the population form a simplex, so that the columns of $S_{k+1}$ span $\mathbb{R}^n$, and (4.12) is positive definite with $\Gamma\Gamma^T = 0$.

In that case, (4.11) becomes

$$B_{k+1} = Y_{k+1}\Omega^2 S_{k+1}^T \left(S_{k+1}\Omega^2 S_{k+1}^T\right)^{-1}. \tag{4.13}$$

If there are exactly $n+1$ iterates forming a simplex, the geometrical approach is equivalent to the interpolation method proposed by Wolfe (1959), and (4.13) is exactly (4.9), as $S_{k+1}$ is square and non singular in that case. This approach has not shown good numerical behavior in practice as mentioned in Section 4.3. Also, it requires at least $n+1$ iterates, and may not be appropriate for large-scale problems.

The *subspace decomposition* approach is based on the QR decomposition of $S_{k+1}$. We denote by $r$ the rank of $S_{k+1}$, with $r \leq n$, and we have $S_{k+1} = QR$, where

$$Q = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \tag{4.14}$$

with $Q_1$ is $(n \times r)$, $Q_2$ is $(n \times n - r)$, and $R$ is $(n \times k+1)$. The $r$ columns of $Q_1$ form an orthogonal basis of the range of $S_{k+1}$. We define now $\Gamma$ such that

$$\Gamma = \begin{pmatrix} 0_{n\times r} & Q_2 \end{pmatrix} \tag{4.15}$$

that is $Q$ where $Q_1$ has been replaced by a null matrix. With this construction $\Gamma\Gamma^T + S_{k+1}\Omega^2 S_{k+1}^T$ is invertible and $S_{k+1}\Gamma\Gamma^T = 0$. In the case where $S_{k+1}$ spans the entire space then $r = n$, $\Gamma$ is a null matrix and (4.11) is equivalent to (4.13).

With the subspace decomposition approach, the changes of $F$ predicted by $B_{k+1}$ in a direction orthogonal to the range of $S_{k+1}$ is the same as the one predicted by the arbitrary matrix $B_{k+1}^0$. This idea is exactly the same as the one used by Broyden (1965) to construct his so called *Broyden's Good method.*

Numerical problems may occur when the columns of $S_{k+1}$ are close to linear dependence. These are the problems already mentioned in the introduction, and reported namely by Ortega and Rheinboldt (1970) and Dennis and Schnabel (1996). Clearly, such problems do not occur when $S_{k+1}$ has exactly one column, which leads to the classical Broyden method.

The *numerical approach* is designed to address both the problem of overcoming the under-determination, and guaranteeing numerical stability. It is directly inspired by the modified Cholesky factorization proposed by Schnabel and Eskow (1991). The modified Cholesky factorization of a square matrix $A$ creates a matrix $E$ such that $A + E$ is safely positive definite, while computing its Cholesky factorization. It may namely happen that $A$ has full rank, but with smallest eigenvalue very small with regard to machine precision. In that case, $E$ is non zero despite the fact that $A$ is non singular. We apply this technique with $A = S_{k+1}\Omega^2 S_{k+1}^T$ and $E = \Gamma\Gamma^T$. So, if the matrix $S_{k+1}\Omega^2 S_{k+1}^T$ is safely positive definite, $\Gamma\Gamma^T = 0$ and (4.11) reduces to (4.13). If not, the modified Cholesky factorization guarantees that the role of the arbitrary term $\Gamma$ is minimal.

We now emphasize important advantages of our generalization combined with the *numerical approach*. Firstly, contrarily to interpolation methods, our least squares model allows to use more than $p$ points to identify a model in a subspace of dimension $p$ (where $p \leq n$). This is very important when the objective function is expensive to evaluate. Indeed, we make an efficient use of all the available information about the function to calibrate the secant model. It is namely advantageous compared to Broyden's method, where only two iterates are explicitly used to build the model, while previous iterates only play an implicit role due to the "least-change" principle. Secondly, the numerical approach proposed above controls the numerical stability of the model construction process, when a sequence of iterates is (almost) linearly dependent. Finally, the fact that existing methods are special cases of our approach allows to generalize the theoretical and practical properties already published in the literature, and simplifies their extension to our context. The main drawback is the increase in numerical linear algebra as the least squares problem (4.10) must be solved at each iteration. Therefore, it is particularly appropriate for problems where $F$ is very expensive to compute.

We conclude this section by showing that our population-based update formula is a generalization of Broyden update. Actually, the classical Broyden update (4.6) is a special case of our update formula (4.11), if $B^0_{k+1} = B_k$, the population contains just two iterates $x_k$ and $x_{k+1}$, and the *subspace decomposition* approach is used. The secant equation (4.5) completely defines the linear model in the one-dimensional subspace spanned by $s_k = x_{k+1} - x_k$, while an arbitrary decision is made for the rest of the model. If we define $\omega^k_{k+1} = 1$ and $\Gamma$ is given by (4.15) with $r = 1$, we can write (4.11) as

$$B_{k+1} = B_k + (y_k - B_k s_k)\, s_k^T \left(\Gamma\Gamma^T + s_k s_k^T\right)^{-1}. \tag{4.16}$$

The equivalence with (4.6) is due to the following equality

$$s_k^T \left(\Gamma\Gamma^T + s_k s_k^T\right)^{-1} = s_k^T \frac{1}{s_k^T s_k}, \tag{4.17}$$

obtained from the fact that $s_k^T \Gamma\Gamma^T = 0$, by (4.15).

Bierlaire et al. (to appear) have shown that if $\Gamma\Gamma^T$ is determined by the *numerical approach* described above, then the undamped algorithm described in Section 4.4, where $B_{k+1}$ is defined by (4.11) in its update form (*i.e.* $B^0_{k+1} = B_k$), locally converges to a solution of (4.1) under standard assumptions.

## 4.4 Undamped and damped quasi-Newton methods

All the algorithms presented in Section 4.3.1 and 4.3.2 are based on the following structure.

- Given $F : \mathbb{R}^n \to \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$, $B_0 \in \mathbb{R}^{n \times n}$ and $k = 0$
- While a stopping criterion is not verified:
  - Find $s$ solving $B_k s = -F(x_k)$,
  - Evaluate $F(x_{k+1})$ where $x_{k+1} = x_k + s$,
  - Compute $B_{k+1}$;
  - Set $k = k + 1$.

This general algorithm is often called *undamped* quasi-Newton method, *i.e.* without any step control or globalization methods. It allows to compare different types of algorithms, in terms of number of function evaluations, and their robustness without introducing a bias due to the step control or the globalization method. Consequently, the algorithms differ only by the method used to compute $B_{k+1}$.

Newton's method and other Newton-like methods (such as quasi-Newton methods) are known to be an efficient way for solving (4.1). For example, Newton's method is known to exhibit a quadratic rate of convergence to a solution (root) $x^*$ of $F$ provided that the starting point $x_0$ is sufficiently close to $x^*$.

However an undamped Newton-based method, taking unit step lengths at each iteration, may fail to converge to a solution of the system due to the lack of control on the step taken at each iteration of the algorithm. The main drawback of undamped Newtonian methods is that we cannot ensure convergence from remote starting points. Indeed, in order to guarantee convergence to a solution of the system, the starting point has to be in the vicinity of this solution. Moreover, Newton-like methods without any control on the step length may encounter several other sources of failure. For instance, the components of the unknown vector $(x)$ or the function vector $(F)$ or the Jacobian approximate $(B_k)$ may become arbitrarily large. Finally, the algorithm can sometimes cycle between two iterates or two distinct regions of the search space. In conclusion, the robustness of undamped quasi-Newton methods can be poor when solving difficult problems.

Secant methods, and our method in particular, can be made more robust. Indeed, the use of a globalization technique allows to considerably improve the robustness of Newtonian methods, ensuring convergence even from remote starting points.

Most globalization techniques can be grouped into two distinct frameworks, each coming from nonlinear optimization research community. On the one hand, linesearch techniques construct a merit function based on $F$ in order to measure progress toward a solution of the problem. A step length is typically accepted only if it provides a sufficient decrase in F. On the other hand, trust-region and filter-trust-region methods consist in optimization algorithms and are applied to the associated least-squares problem, that is:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|F(x)\|_2^2.$$

The drawback of this approach is that iterates may get stuck in a local minimum of the optimization problem, ending the optimization process without getting a solution of the original system.

Our globalization techniques are different in the sense that we consider this sum-of-squares function only as a merit function to measure progress toward a solution while we keep solving the original problem, using a linesearch framework first and then a linesearch-filter framework.

When integrating a globalization strategy to the previous undamped quasi-Newton framework, we obtain the following structure.

- Given $F : \mathbb{R}^n \to \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$, $B_0 \in \mathbb{R}^{n \times n}$ and $k = 0$
- While a stopping criterion is not verified:
  - Find $s$ solving $B_k s = -F(x_k)$;
  - Determine a step length $\alpha_k > 0$;
  - Evaluate $F(x_{k+1})$ where $x_{k+1} = x_k + \alpha_k s$;
  - Compute $B_{k+1}$;
  - Set $k = k + 1$.

This general method is called *damped* quasi-Newton method.

In the following, we describe how we determine the step $\alpha_k$ using the two globalization strategies we propose. In both strategies, we use at each iteration the classical sum-of-squares merit function

$$m(x_k) = \frac{1}{2} \|F(x_k)\|_2^2 = \frac{1}{2} \sum_{i=1}^{n} F_i^2(x_k) \tag{4.18}$$

to measure progress toward a solution of the system F.

Note that $s$ may not be chosen as search direction when a reduction in (4.18) cannot be obtained along this direction.

### 4.4.1 Linesearch approach

In this globalization strategy, similarly to inexact linesearches, we choose a step $\alpha_k$ satisfying the following Armijo-type condition with $\beta \in (0, 1)$:

$$m(x_k + \alpha_k s) \leq m(x_k) + \alpha_k \beta \nabla m(x_k)^T s. \tag{4.19}$$

Note that $\beta$ is a parameter which defines the quality of the decrease we want to obtain. Condition (4.19) is valid only if the quasi-Newton direction $s$ is a descent direction for $m$ in $x_k$, that is:

$$\nabla m(x_k)^T s < 0. \tag{4.20}$$

If condition (4.20) holds, we find a step $\alpha_k$ satisfying (4.19) using a backtracking strategy. Unfortunately, we do not have the guarantee that our quasi-Newton direction $s = -B_k^{-1}F(x_k)$ is a descent direction for $m$, unless $B_k$ is close enough to the real Jacobian at $x_k$, $J(x_k) = \nabla F(x_k)^T$, and $\nabla m(x_k)^T s$ is bounded below. Consequently, we use the following sequential procedure to find a descent direction for the merit function in the current iterate $x_k$:

- Check whether the quasi-Newton direction $s = -B_k^{-1}F(x_k)$ is a descent direction for $m$ in $x_k$;

- If not, compute using the modified Cholesky factorization (see Schnabel and Eskow, 1999) an auxiliary direction $\bar{s}$

  $$-(B_k^T B_k + \tau I)^{-1} B_k^T F(x_k) \tag{4.21}$$

  where $\tau > 0$ and $I$ is the identity matrix in dimension $n$. According to Nocedal and Wright (1999), we can always choose $\tau$ to ensure that $\nabla m(x_k)^T s$ is bounded below.

- Check whether the quasi-Newton direction $\bar{s}$ is a descent direction for $m$ in $x_k$;

- If not, do the following:

  – Update the current approximation of the Jacobian $B_k$ with a new point close to $x_k$ to get $B_k^+$. More precisely, we take a step of length $10^{-4}$ in the direction $s$. The goal is to try to get a good local approximation of $J(x_k)$;

  – Compute the direction $s^+ = -(B_k^+)^{-1}F(x_k)$;

  and restart the process with $s^+$.

Note that we compute the directional derivative of the merit function $m$ in a direction $s$, $\nabla m(x)^T s$, using a finite differences procedure. Thus, it may happen that the numerical approximation does not have the same sign as the analytical value.

### 4.4.2 Linesearch-filter approach

#### 4.4.2.1 Filter methods

The question of using a filter method to solve systems of equations has been first addressed by Fletcher and Leyffer (2003).

The original filter proposed by Fletcher and Leyffer (2002) had two dimensions associated with the two conflicting goals of constrained optimization, namely satisfy the constraints and minimize the objective function. Multidimensional filters have been proposed by Gould et al. (2005) for solving systems of nonlinear equations and nonlinear least-squares and consequently by Gould et al. (2006) in the context of unconstrained optimization.

Originally used in a trust-region framework, filter methods have also been combined with linesearch techniques in the context of constrained optimization, using successfully the increased flexibility of the filter concept. Wächter and Biegler (2001) have developed a unified filter framework based on linesearch which can be adapted to interior-points methods as well as SQP methods (see also Wächter and Biegler, 2005b and Wächter and Biegler, 2005a for convergence results). Benson et al. (2002) have presented a way to implement the filter concept inside their interior points algorithm LOQO.

We propose here a new globalization technique for solving systems of nonlinear equations, that is, a multidimensional filter method combined with a linesearch approach. This algorithm is based on the linesearch framework presented above within which we add the filter concept as a second criterion to accept iterates and to assess the quality of a trial step length $\alpha_k$. We start by defining the filter and then we describe the main steps of the linesearch-filter algorithm in order to compute the step length.

As we are interested in solving the original problem

$$F(x) = 0$$

we consider violations of single equations $F_j(x) = 0$, that is $|F_j(x)|$, as quantities to measure progress toward a solution of the system. Consequently we define

$$\theta_{j,k} = \theta_j(x_k) = |F_j(x_k)| \quad \forall j = 1, \ldots, n.$$

We say that an iterate $x_k$ is dominated by an iterate $x_l$ if

$$\theta_j(x_l) \leq \theta_j(x_k) \quad \forall j = 1, \ldots, n. \tag{4.22}$$

Consequently, we consider that $x_k$ is of no interest if we keep $x_l$. Indeed, $x_l$ is at least as good as $x_k$ for each equation of the system.

Given the concept of dominance defined above, the filter, which is called $\mathcal{F}$, remembers all non-dominated iterates during the optimization process. In other words, the filter $\mathcal{F}$ is a list of $n$-tuples $(\theta_{1,k}, \dots, \theta_{n,k})$ such that

$$\theta_{j,k} < \theta_{j,l} \text{ for at least one } j \in \{1, \dots, n\}$$

for $k \neq l$.

Filter methods accept a trial iterate $x_k^+ = x_k + \alpha_k s$ if it is not dominated by any other point already in the filter $\mathcal{F}$. However, in order to make the filter concept efficient from an algorithmic point of view, we slightly strengthen the acceptance condition for the filter by using the idea of sufficient improvement widely used in nonlinear optimization. More formally, we say that an iterate $x_k^+$ is *acceptable for the filter* $\mathcal{F}$ if and only if

$$\forall \theta_l \in \mathcal{F}\ \exists j \in \{1, \dots, n\} \quad \theta_j(x_k^+) < \theta_{j,l} - \gamma_\theta \|\theta_k^+\| \tag{4.23}$$

where $\gamma_\theta$ is a positive constant, small enough such that the right-hand side of (4.23) remains positive. This condition is necessary to guarantee that points acceptable for the filter always exist. Not only we do impose a sufficient improvement but the required improvement is proportionnal to the "global" violation of the trial iterate $\|\theta_k^+\|$.

Note that the fact of adding a new entry $\theta_k$ to the filter $\mathcal{F}$ may cause other entries in the filter to be dominated by this new entry in the sense of (4.22). We therefore remove an entry $\theta_l$ from the filter if

$$\forall j \in \{1, \dots, n\} \quad \theta_{j,l} \geq \theta_{k,l}$$

In what follows, we present the linesearch-filter technique which aims to globalize quasi-Newton methods for solving systems of nonlinear equations and we refer to (4.23) as the *filter condition*.

#### 4.4.2.2 Linesearch-filter algorithm

As we have seen, (4.23) provides an additional condition to accept a trial step length compared to the linesearch framework where we only use the Armijo-type sufficient decrease condition (4.19) for the merit function (4.18). The filter being more flexible and less conservative than the classical condition in a linesearch framework, it plays the major role in the algorithm. We fall back on the previous linesearch framework when things are going badly.

We keep using a backtracking strategy to compute the sequence of steps to be tested. For each trial step length, we first test the filter condition. If it does not hold, we test the sufficient decrease condition.

Our linesearch-filter approach to compute $\alpha_k$ is summarized by the following sequential procedure. Note that, contrarily to the linesearch framework, we have a non-monotone algorithm in the sense that it is not mandatory that the condition (4.20) holds at each iteration of the algorithm. Indeed, a trial point $x_k^+ = x_k + \alpha_k s$ may satisfy the filter condition (4.23) even if $s$ is not a descent direction for the merit function.

- Check whether the quasi-Newton direction $s = -B_k^{-1}F(x_k)$ is a descent direction for the merit function $m$ in $x_k$;
  - If it is, check consecutively the filter condition (4.23) and the sufficient decrease condition (4.19) for each trial step length $\alpha_k$ obtained by the backtracking strategy.
  - If not, check the filter condition (4.23) for `maxiter` trial step lengths.
- If `maxiter` is reached, compute the auxiliary direction $\bar{s}$ given by (4.21) and check whether condition (4.20) holds;
  - If it holds, check consecutively the filter condition (4.23) and the sufficient decrease condition (4.19) for each trial step length $\alpha_k$ obtained by the backtracking strategy.
  - If not, check the filter condition (4.23) for `maxiter` trial step lengths.
- If `maxiter` is again reached,
  - Update the current quasi-Newton approximation $B_k$ as in the linesearch framework to get $B_k^+$;
  - Compute the new direction $s^+ = -(B_k^+)^{-1}F(x_k)$;
  - Restart the process with $s^+$.

Note that $\nabla m(x)^T s$ is still computed using a finite differences procedure. Note also that `maxiter` is set to a small value, typically 3. It means that the algorithm only tests a couple of large step lengths whose corresponding trial iterate might be acceptable for the filter, even if the associated search direction does not give rise to a reduction in the merit function. The idea is again to give more flexibility in accepting iterates as it represents another possibility to take large steps toward a solution. But in any case, we do not want to take a small step when the direction in which we search is not a descent direction for the merit function (4.18).

## 4.5 Numerical experiments

We present here an analysis of the performance of our method, in comparison to classical algorithms. All algorithms and test functions have been implemented with the package Octave (Eaton, 1997a) and computations have been done on a desktop equipped with 3GHz CPU in double precision. The machine epsilon is about $2.2\ 10^{-16}$.

The numerical experiments were carried out on a set of 43 test functions. For 37 of them, we consider five instances of dimension $n = 6, 10, 20, 50, 100$. We obtain a total of 191 problems. This set is composed of the four standard nonlinear systems of equations proposed by Dennis and Schnabel (1996) (that is, *Extended Rosenbrock Function*, *Extended Powell Singular Function*, *Trigonometric Function*, *Helical Valley Function*), three functions from Broyden (1965), five functions proposed by Kelley (2003) in his book on Newton's method (that is, *Arctangent Function*, *a Simple Two-dimensional Function*, *Chandrasekhar H-equation*, *Ornstein -Zernike Equations*, *Right Preconditioned Convection-Diffusion Equation*), three linear systems of equations (see in Appendix A.2), the test functions given by Spedicato and Huang (1997) and some test functions of the collection proposed by Moré et al. (1981). For each problem, we have used the starting point proposed in the original paper. Note that the results include all these problems.

The algorithms are based on both the damped and undamped quasi-Newton framework given in Section 4.4 with the following characteristics: the initial Jacobian approximation $B_0$ is the same for all algorithms and equal to the identity matrix. The stopping criterion is a composition of three conditions: small residual, that is $\|F(x_k)\|/\|F(x_0)\| \leq 10^{-6}$, maximum number of iterations ($k \geq 200$ for problems of size $n \leq 20$ and $k \geq 500$ for problems of size $n > 20$), and divergence, diagnosed if $\|F(x_k)\| \geq 10^{10}$ or if a descent direction has not been found after several updates of the approximate Jacobian in the linesearch procedure (meaning that we have not been able to find a sufficiently good approximation of the Jacobian).

We consider four quasi-Newton methods:

1. Broyden's Good Method (BGM), using the update (4.6).
2. Broyden's Bad Method (BBM), also proposed by Broyden (1965). It is based on the following secant equation:

$$s_k = B_{k+1}^{-1} y_k. \tag{4.24}$$

   and directly computes the inverse of $B_k$:

$$B_{k+1}^{-1} = B_k^{-1} + \frac{\left(s_k - B_k^{-1} y_k\right) y_k^T}{y_k^T y_k}. \tag{4.25}$$

Broyden (1965) describes this method as "bad", that is numerically unstable. However, we have decided to include it in our tests for the sake of completeness. Moreover, as discussed below, it does not always deserve its name.

3. The Hybrid Method (HMM) proposed by Martínez and Ochi (1982). At each iteration, the algorithm decides to apply either BGM or BBM. Martínez (2000) observes a systematic improvement of the Hybrid approach with respect to each individual approach. As discussed below, we reach similar conclusions.

4. Our population-based approach, called Generalized Secant Method (GSM), defined by (4.11) in its update form with $B^0_{k+1} = B_k$ using the *numerical approach* described in Section 4.3.2, with $\tau = (\text{macheps})^{\frac{1}{3}}$ and a maximum of $p = \max(n, 10)$ previous iterates in the population. Indeed, including all previous iterates, as proposed in the theoretical analysis, may generate memory management problems, and anyway does not significantly affect the behavior of the algorithm. The weights are defined as

$$\omega^i_{k+1} = \frac{1}{\|x_{k+1} - x_i\|^2} \qquad \forall i \in I_p \tag{4.26}$$

The measure of performance is the number of function evaluations to reach convergence. Indeed we are interested in applying the method on computationnally expensive systems, where the running time is dominated by the function evaluations. We are presenting the results following the performance profiles analysis method proposed by Dolan and Moré (2002) and described in Section 2.6.2. In this case, the performance index is the number of function evaluations.

We first analyze the performance profile of all algorithms described above without globalization strategy on all problems. The performance profile is reported on Figure 4.1. A zoom on $\pi$ between 1 and 5 is provided in Figure 4.2.

The results are very satisfactory for our method. Indeed, we observe that GSM is the most efficient and the most robust algorithm among the challenged quasi-Newton methods.

We also confirm results by Martínez (2000) showing that the Hybrid method is more reliable than BGM and BBM. Indeed, it converges on almost 50% of the problems, while each Broyden method converges only on less than 40% of the cases. Moreover, HMM wins more often than BGM and BBM does, and is also more robust, as its performance profile grows faster than the profile for BGM and BBM. The relative robustness of BGM and BBM is comparable.

Even if GSM is the most reliable algorithm, note that it only converges on 55% of the 191 runs. We now present the performance profile for all algorithms in their damped ver-

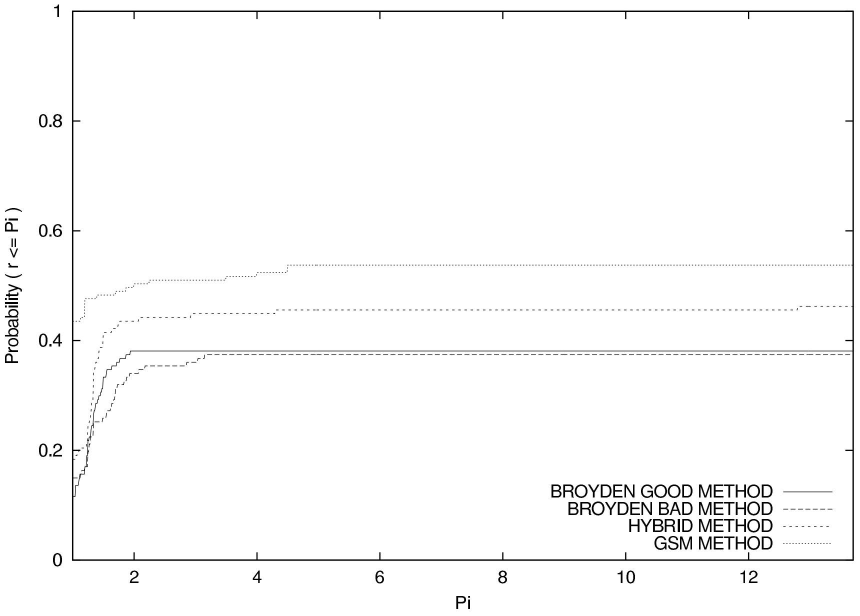


Figure 4.1: Performance Profile without globalization

sion, that is making use of the linesearch strategy presented in Section 4.4, on Figure 4.3. A zoom for $\pi$ between 1 and 3 is provided in Figure 4.4. Firstly we observe that the globalization technique significantly improves the robustness of all four presented algorithms as expected. Secondly and most importantly, GSM remains the best algorithm in terms of efficiency and robustness. More precisely, GSM is the best algorithm on more than 60% of the problems and is able to solve more than 80% of the 191 considered problems. From Figure 4.4, we note also that when GSM is not the best method, it converges within a factor of 2 of the best algorithm for most problems.

The performance profile analysis depends on the number of methods that are being compared. Therefore, we would like to present a comparison between BGM and GSM only, as BGM is probably the most widely used method. The significant improvement provided by our method over Broyden's method is illustrated by Figure 4.5 considering the undamped version of both algorithms. Figure 4.6 shows the superiority of GSM as well, when both algorithms are globalized using the linesearch strategy.

In this chapter, in the context of solving systems of nonlinear equations, we focused on quasi-Newton methods which do not use information about the derivative of the system to be solved. We have already shown that GSM is a very competitive derivative-free algorithm. To conclude our numerical experiments, we would like to compare our method with an algorithm using derivative information.

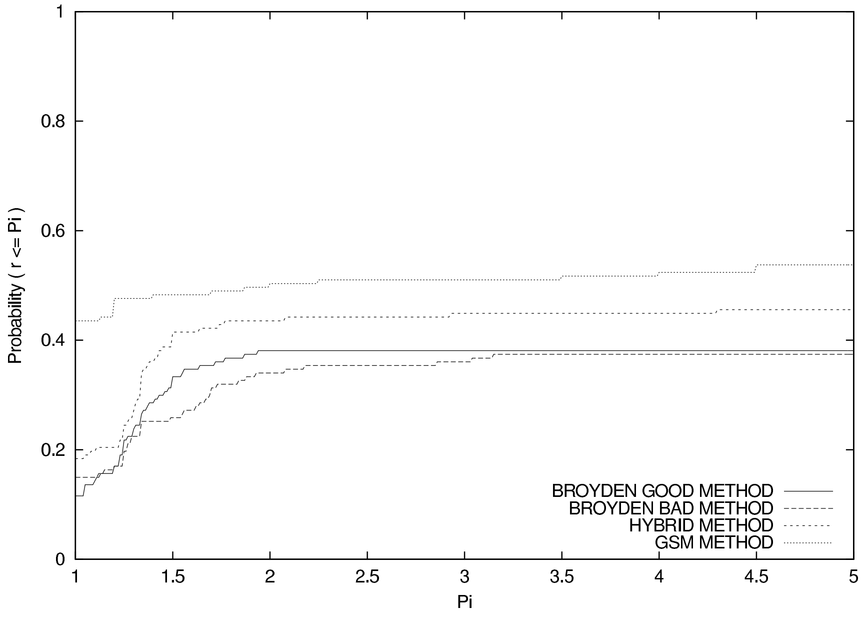


Figure 4.2: Performance Profile on (1,5) without globalization

We consider a method belonging to the family of inexact Newton methods which identify a direction $d_k$ satisfying the inexact Newton condition:

$$\|F(x_k) + J(x_k)d_k\| \leq \eta_k \|F(x_k)\| \tag{4.27}$$

for some $\eta_k \in [0, 1)$. The most conventional inexact Newton method uses iterative techniques to compute the Newton step $d_k$ using (4.27) as a stopping criterion. Among these iteratives techniques, Krylov-based linear solvers are generally chosen. Newton-Krylov methods need to estimate Jacobian-vector products using finite differences approximations in the appropriate Krylov subspace.

We now challenge GSM against the Newton-Krylov method presented by Kelley (2003). The considered version of this method uses the iterative linear GMRES (proposed by Saad and Schultz, 1986) and a parabolic linesearch via three interpolation points. Similarly to the Newton-Krylov algorithm, we allow GSM to use a finite differences approximation of the initial Jacobian. From Figure 4.7, we observe that GSM is competitive with Newton-Krylov both in terms of efficiency and robustness. This result is very satisfactory as Newton-Krylov methods have been proven to be very efficient methods to solve systems of nonlinear equations.

We now present the performance profiles of all 4 quasi-Newton algorithms in their damped version using the linesearch-filter framework in Figure 4.8. A zoom for $\pi$ between 1 and 5 is provided in Figure 4.9. From Figure 4.9, we see that GSM is the best method

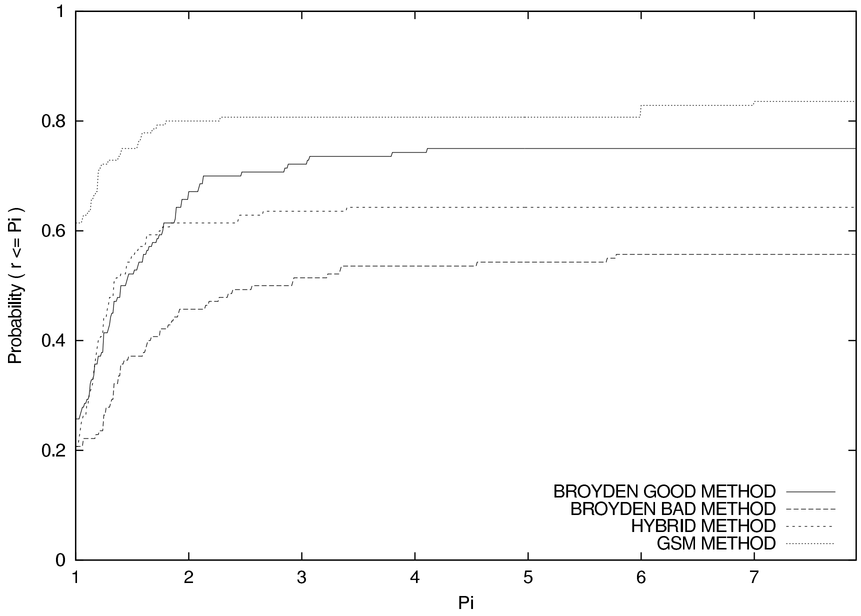


Figure 4.3: Performance Profile with linesearch

on about 55% of the problems. One can see that the robustness of quasi-Newton methods obtained with this globalization technique is slightly less than the robustness attained with the linesearch strategy. However, GSM is still able to solve 75% of the tested problems, as it can be seen on Figure 4.8.

We will see later that the linesearch-filter framework exhibits best results in terms of robustness, when starting the algorithm with a finite-differences approximation of the Jacobian. It is mainly due to the fact that the filter is less conservative. Consequently, when starting the optimization process with the identity matrix as Jacobian approximate, the linesearch-filter strategy accepts iterates very easily during the first iterations of the algorithm and chosen step lenghts may provide poor subsequent iterates, causing the algorithm to fail to converge.

From these two profiles, we can also note that the Hybrid Method behaves better compared to both Broyden-based methods in terms of robustness and efficiency. This superiority of the Hybrid Method when globalized with our linesearch-filter approach is consistent with the respective performances of these quasi-Newton methods in their undamped version. The explanation can be found in the very motivation of using a filter technique. Remember that the idea of filter method is too interfere as little as possible with Newton's method whose behaviour is locally very good while ensuring it to be globally convergent. We can see from these profiles that it is also true when we combine the filter

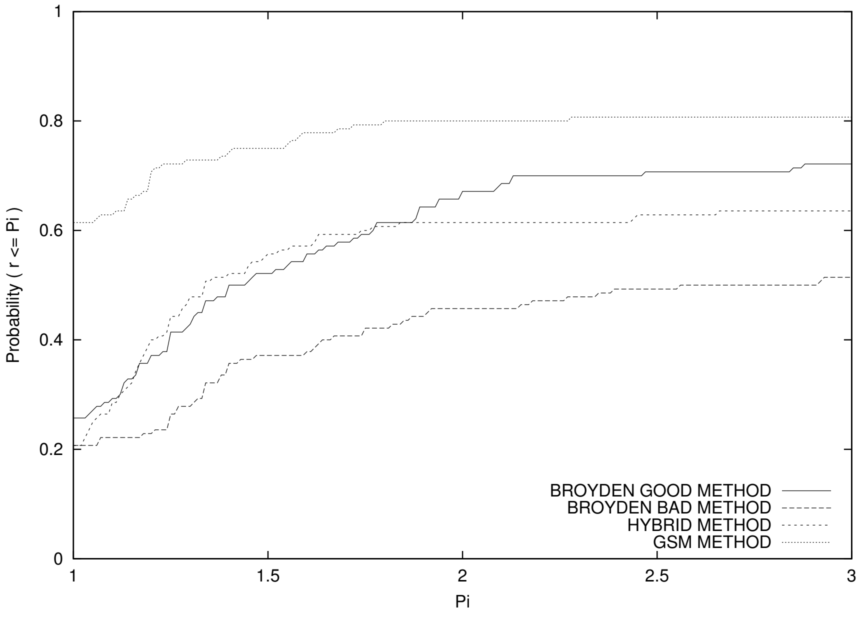


Figure 4.4: Performance Profile on (1,3) with linesearch

technique with quasi-Newtom algorithms. The ranking of these algorithms is kept the same when globalizing all of them with the linesearch-filter scheme, while they are all made more robust.

When using linesearch to globalize the four same algorithms, the Broyden's Good Method clearly outperforms in efficiency and robustness the Hybrid Method. It seems to be particularly well adapted for a linesearch framework.

As performance profiles depend on the number of algorithms that are compared, we propose in Figures 4.10(a) and 4.10(b) a more specific analysis of the performance of GSM compared to Broyden's Good Method, which is probably the most widely used quasi-Newton method for systems of nonlinear equations. From these profiles, we can see that GSM significantly outperforms BGM. GSM is able to solve an additional significant percentage of the problems we have tested, about 15%. Also, it is the most efficient algorithm in terms of number of function evaluations on around 60% of the test functions.

As we did with GSM method using linesearch, we now compare GSM method combined with linesearch-filter technique with the Newton-Krylov method. As we are comparing our derivative-free method with a method using information on derivatives by the mean of finite differences, we allow GSM to use a finite-differences approximation of the Jacobian at the initial iterate $x_0$.

Results are shown on Figures 4.11(a) and 4.11(b). There are even more satisfactory

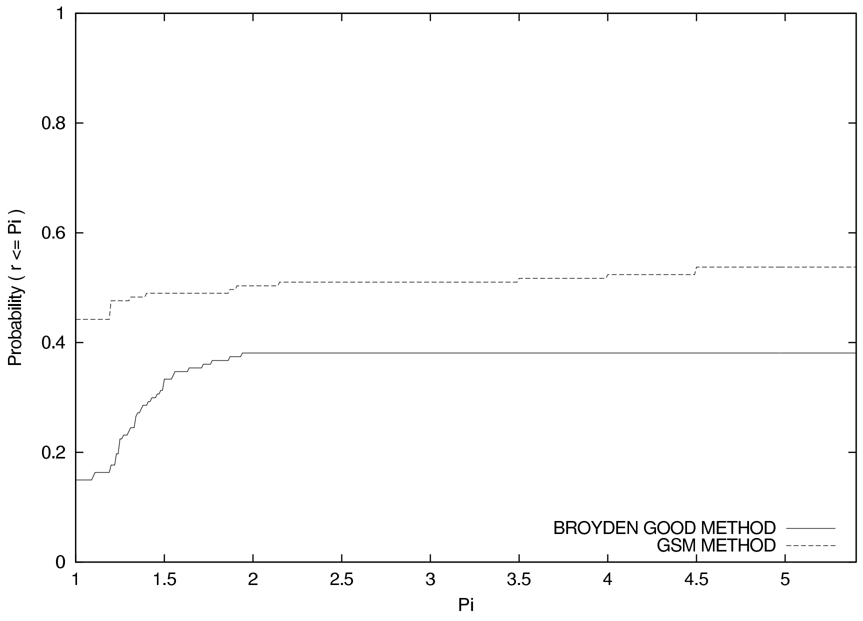


Figure 4.5: Performance profile without globalization – Broyden's Good Method and GSM

than the results obtained when comparing GSM with linesearch and Newton-Krylov. As we already discussed it, the fact of using a good starting approximation for the Jacobian allows to avoid failure of the linesearch-filter method during the first iterations. GSM is nearly as competitive as Newton-Krylov, as it is the best algorithm on nearly 50% of the problems. Moreover, from Figure 4.11(b), we see that GSM is either the best method or within a factor 2 of the other method on around 65% of the problems, while for Newton-Krylov it is only the case on 50% of the problems. Finally, GSM is a bit more robust since the overall probability of solving a problem (0.85) is higher compared to Newton-Krylov (0.80).

We now compare in Figures 4.12(a) and 4.12(b) both globalized GSM methods with Newton-Krylov method. Again we note that both GSM-based algorithms are more robust than Newton-Krylov while being competitive in terms of number of function evaluations to reach convergence. Among the three challenged algorithms, GSM with linesearch-filter is still either the best algorithm or within a factor 2 of the best one on 65% of the problems. Finally, note that both globalization techniques gives the same level of robustness to GSM update when a finite-differences approximation of the Jacobian is used.

Before concluding these numerical tests, we now propose to only compare both globalization approaches of GSM method. Figures 4.13(a) and 4.13(b) compare both algorithms when $H_0$ is given by the identity matrix while figures 4.14(a) and 4.14(b) compare the

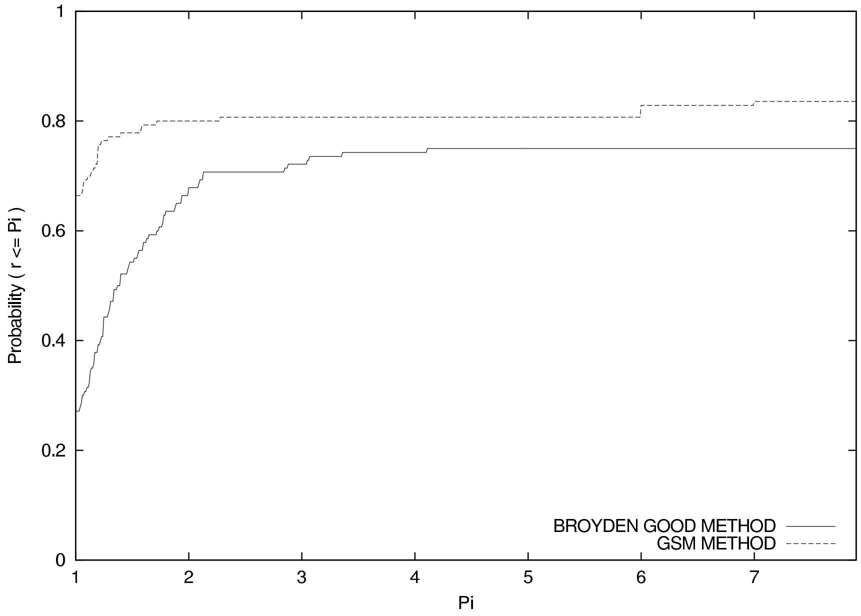


Figure 4.6: Performance profile with linesearch – Broyden's Good Method and GSM –

same algorithms when the initial Jacobian is obtained by finite-differences. Note that, on these 4 profiles, the sum of probabilities of being the best algorithm is greater than 1. It is due to the fact that both methods behave exactly the same on some problems, which could be expected as the linesearch-filter integrates the linesearch one. While the robustness of GSM with linesearch seems to be somehow independent from the choice of the initial Jacobian, one can see that the quality of this initial Jacobian plays a role on the efficiency and the robustness of the linesearch-filter variant. Indeed, while the linesearch variant is significantly better than the linesearch-filter variant on Figure 4.13(b), 4.13(b), both variants have essentially the same level of performance on Figure 4.14(b).

### 4.5.1 Large-scale problems

The main drawback of our approach is the relatively high cost in numerical linear algebra. Therefore it is particularly appropriate for medium-scale problems where $F$ is very expensive to compute. Bierlaire and Crittin (2006) propose an instance of this class of methods, designed to solve very large-scale systems of nonlinear equations without any assumption about the structure of the problem. The numerical experiments on standard large-scale problems show similar results: the algorithm outperforms classical large-scale quasi-Newton methods in terms of efficiency and robustness and its numerical performances are similar to Newton-Krylov methods. Moreover, our algorithm is robust in

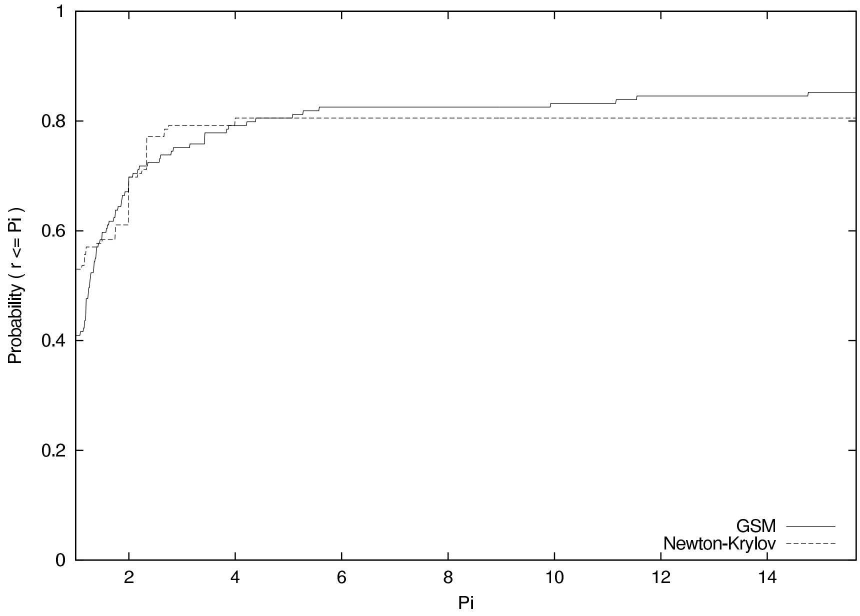


Figure 4.7: Performance profile – GSM with linesearch and Newton-Krylov –

presence of noise.

The complexity (both in time and memory) is linear in the size of the problem. Therefore, we were able to solve very large instances of a problem given by Spedicato and Huang (1997). The algorithm has been able to converge on a problem of size 2'000'000 in four hours and 158 iterations.

We are strongly interested in globalizing the large-scale version of our method. However, it requires future research to adapt our linesearch and linesearch-filter frameworks and to get an efficient globalization strategy in terms of computational time.

## 4.6 Conclusions

We have proposed a new class of generalized secant methods, based on the use of more than two iterates to identify the secant model. Contrarily to previous attempts for multi-iterate secant methods, the key ideas of this chapter are (i) to use a least squares approach instead of an interpolation method to derive the secant model, and (ii) to explicitly control the numerical stability of the method.

A specific sub-class of this family of methods provides an update formula. Moreover, we have performed extensive numerical experiments with several algorithms. The results show that our method produces significant improvement in terms of robustness and number

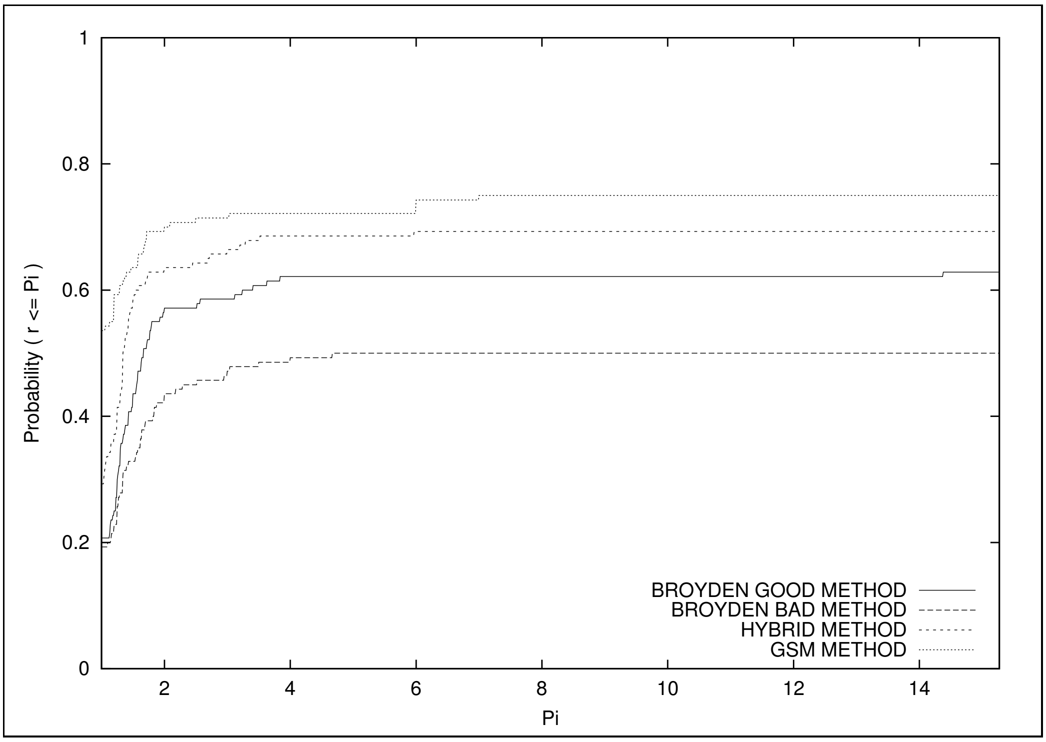


Figure 4.8: Performance profile with linesearch-filter

of function evaluations compared to classical methods. We have also shown that the globalization strategies presented in this chapter significantly improves the robustness of quasi-Newton methods. Finally, damped versions of GSM have been proved to be competitive derivative-free alternatives to Newton-Krylov methods.

The ability of GSM to significantly reduce the number of function evaluations necessary to solve a system of nonlinear equations makes it particularly appealing for real applications, and in particular transportation contexts, as the systems involved to be solved have the particularity that $F$ is expensive to evaluate.

## 4.7 Perspectives

A theoretical analysis of a globally convergent version of our method must be performed. We also conjecture that the algorithm can be tailored in order to enforce a superlinear rate of local convergence. One possible way is to design the parameters such that the method becomes asymptotically equivalent to Broyden's method and, consequently, would inherit its superlinear rate of convergence. This needs to be investigated further.

There are several variants of our methods that we plan to analyze in the future. Firstly, following Broyden's idea to derive BBM from (4.24), an update formula for $B_{k+1}^{-1}$ can easily

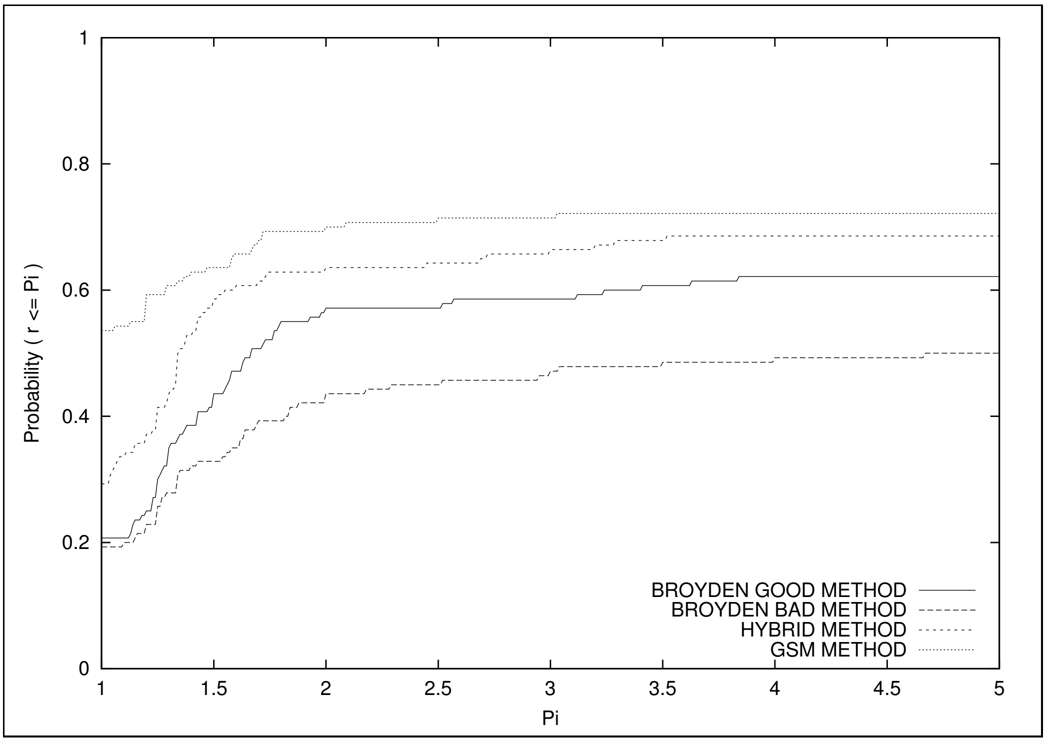


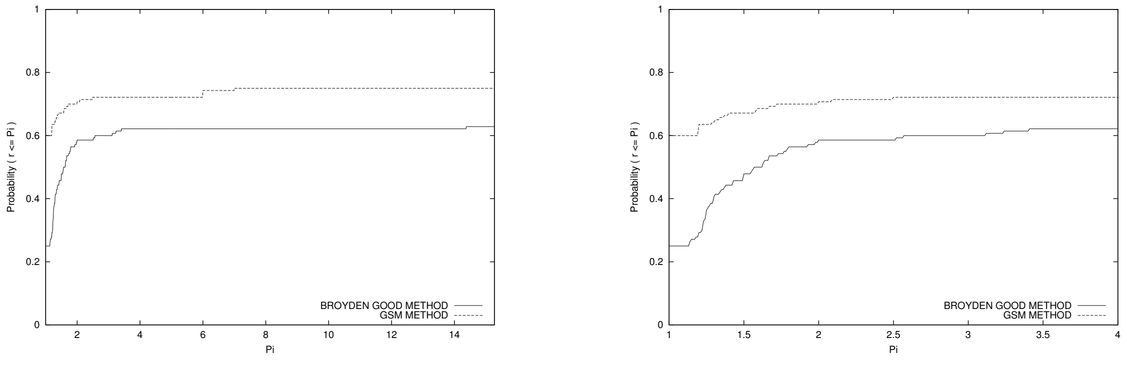
Figure 4.9: Performance profile on (1,5) with linesearch-filter

be derived in the context of our method:

$$B_{k+1}^{-1} = B_k^{-1} + \left(\Gamma\Gamma^T + Y_{k+1}\Omega^2 Y_{k+1}^T\right)^{-1} Y_{k+1}^T \Omega^2 \left(S_{k+1} - B_k^{-1} Y_{k+1}\right). \qquad (4.28)$$

From preliminary tests that we have performed, the "Good" and "Bad" versions of our method compare in a similar way as BGM and BBM. Secondly, non-update instances of our class of methods can be considered. In that case, the arbitrary matrix $B_{k+1}^0$ in (4.10) may be different from $B_k$. Choosing a matrix independent from $k$ allows to use iterative scheme designed to solve large-scale least-squares. In that case, choosing a matrix independent from $k$ would allow to apply Kalman filtering (Kalman, 1960) to incrementally solve (4.10) and, consequently, improve the numerical efficiency of the method. For large scale problems, an iterative scheme such as LSQR (Paige and Saunders, 1982) can be considered. LSQR can also improve the efficiency of Kalman filter for the incremental algorithm (see Bierlaire and Crittin, 2004).

Finally, the ideas proposed in this chapter can be tailored to unconstrained nonlinear optimization including singular problems. We end this chapter by giving some tracks to adapt the proposed generalized secant method to general unconstrained nonlinear optimization as well as to use it to deal with singularities in unconstrained nonlinear opti-



(a) Full profile

(b) Zoom

Figure 4.10: Performance profile with linesearch-filter – Broyden Good and GSM –

mization.

Classical secant methods for unconstrained nonlinear optimization usually compute the approximation $B_k$ of the second derivatives matrix $\nabla^2 f(x_k)$ by using information on the gradient values of the objective function f at the two previous iterates $x_k$ and $x_{k-1}$. Indeed, they require that the gradient of the quadratic model centered at the current iterate $x_k$:

$$m_k(x; B_k) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k) \tag{4.29}$$

interpolates exactly the gradient of the objective function at $x_k$ and $x_{k-1}$, that is:

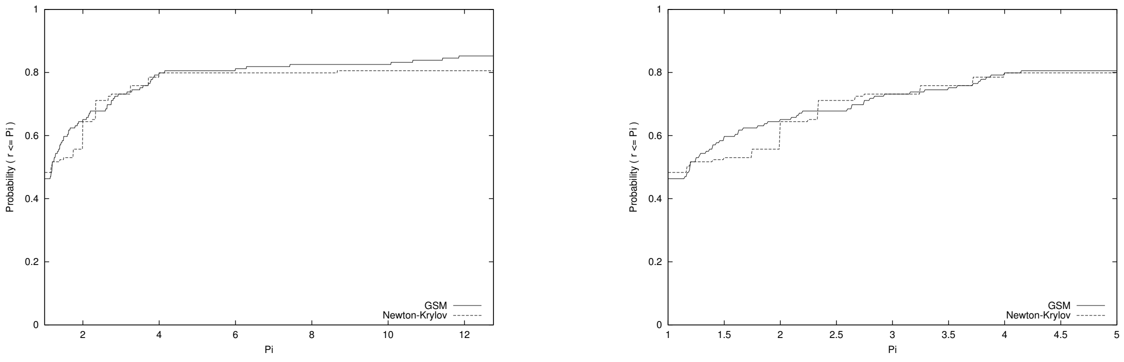
$$\begin{aligned} \nabla m(x_k; B_k) \quad &= \nabla f(x_k), \\ \nabla m(x_{k-1}; B_k) \quad &= \nabla f(x_{k-1}). \end{aligned} \tag{4.30}$$

Subtracting these two equations and defining $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$ and $s_{k-1} = x_k - x_{k-1}$ we obtain the classical secant equation:

$$B_k s_{k-1} = y_{k-1}. \tag{4.31}$$

Clearly, if the number of variables of f is strictly larger than 1, there is an infinite number of matrices satisfying (4.31). Secant methods add consequently some additional requirements on the matrix $B_k$. For example, the BFGS method (see for instance Fletcher, 1970 and Goldfarb, 1970) requires $B_k$ to be symmetric and positive definite and also to be the matrix minimizing variations (in weighted Frobenius norm) between the two successive matrices $B_k$ and $B_{k-1}$.

We present a straightforward adaptation of GSM to solve unconstrained nonlinear optimization problems. At each iteration, we maintain a finite population of previous



(a) Full profile

(b) Zoom

Figure 4.11: Performance profile – GSM with linesearch-filter and Newton-Krylov –

iterates. Without loss of generality, we present the method assuming that all previous iterates $x_0, \dots, x_k$ are considered. In this case, $B_k$ is computed as follows.

$$B_k = \operatorname*{argmin}_{B} \left( \sum_{i=0}^{k-1} \left\| \omega_k^i \nabla f(x_i) - \omega_k^i \nabla m_k(x_i; B) \right\|_2^2 + \left\| B\Gamma - B_k^0 \Gamma \right\|_F^2 \right) \tag{4.32}$$

where $m_k$ is defined by (4.29) and $B_k^0 \in \mathbb{R}^{n \times n}$ is an a priori approximation of $B_k$ (typically we can choose $B_{k-1}$). The role of the second term is still to overcome the underdetermination of the least squares problem based on the first term and also control the numerical stability of the method. The matrix $\Gamma$ contains weights associated with the arbitrary term $B_k^0$, and the weights $\omega_k^i \in \mathbb{R}^+$ are associated with the previous iterates. Equation (4.32) can be written in matrix form as follows:

$$B_k = \operatorname*{argmin}_{B} \left\| B \begin{pmatrix} S_k & I_{n \times n} \end{pmatrix} \begin{pmatrix} \Omega & 0_{k \times n} \\ 0_{n \times k} & \Gamma \end{pmatrix} - \begin{pmatrix} Y_k & B_k^0 \end{pmatrix} \begin{pmatrix} \Omega & 0 \\ 0 & \Gamma \end{pmatrix} \right\|_F^2$$

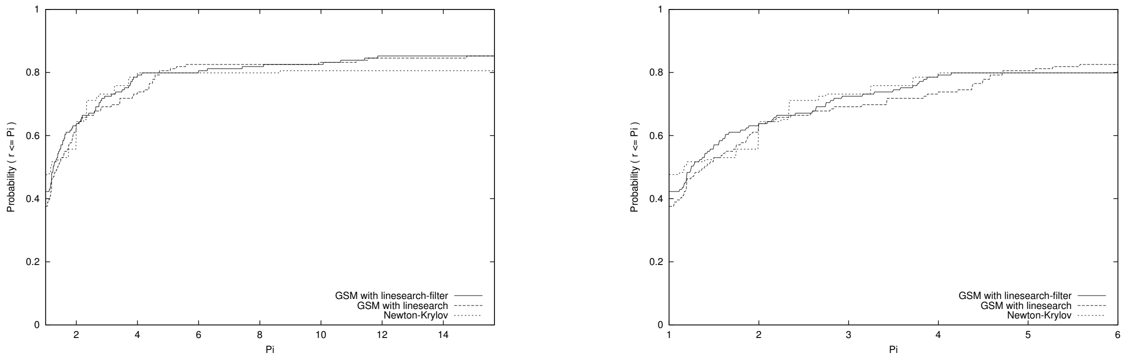
where $\Omega \in \mathbb{R}^k$ is a diagonal matrix with weights $\omega_k^i$ on the diagonal for
$i = 0, \dots, k-1$. The normal equations of this least squares problem lead to the following formula:

$$B_k = B_k^0 + \left( Y_k - B_k^0 S_k \right) \Omega^2 S_k^T \left( \Gamma\Gamma^T + S_k \Omega^2 S_k^T \right)^{-1}, \tag{4.33}$$

where $Y_k = (y_{k-1}, \dots, y_0)$ and $S_k = (s_{k-1}, \dots, s_0)$.

In the case that $B_k^0 = B_k$, (4.33) becomes an update formula. The weights $\omega_k^i$ and $\Gamma$ can be defined as we did in the context of systems of nonlinear equations.

Using this update formula to compute $B_k$ provides an unsymmetric matrix. Invoking the fact that the closest approximation of a symmetric matrix (such as $\nabla^2 f(x_k)$) is itself



(a) Full profile

(b) Zoom

Figure 4.12: Performance profile – GSM with linesearch-filter and linesearch and Newton-Krylov–

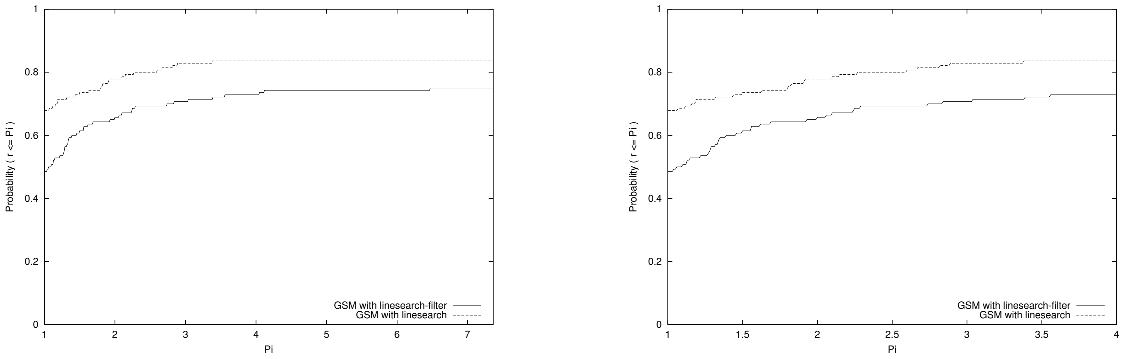
a symmetric matrix, it seems interesting to make our approximation $B_k$ symmetric. We achieve this by projecting the matrix $B_k$ on the space of symmetric matrices. More precisely we compute $B_k^{sym}$ as:

$$B_k^{sym} = \frac{B_k + B_k^T}{2} \tag{4.34}$$

Having a symmetric approximation $B_k$ is also more convenient to compute the derivatives of (4.29). Note that this modification does not represent the best way to make the approximation symmetric. Indeed, when performing the projection, we perturb significantly the approximation obtained by solving the least-squares. We plan to investigate how we could impose the symmetry on the approximation in the least squares formulation in order to get an efficient secant method for unconstrained optimization.

We now present an adaptation of GSM to deal with singularities in unconstrained nonlinear optimization. Inspired by the work of Schnabel and Chow (1991) on tensor methods for singular unconstrained optimization, the idea is to use more information on the objective function to compute the approximation $B_k$. In order to get a more robust method in presence of singularity, our idea is to also use objective function values at previous iterates to calibrate the approximation $B_k$ in the least-squares sense. If we put this additional information in (4.32), we obtain the following least-squares:

$$\begin{aligned} B_k = \operatorname*{argmin}_B \sum_{i=0}^{k-1} \left\| \omega_k^i \nabla f(x_i) - \omega_k^i \nabla m_k(x_i; B) \right\|_2^2 \\ + \sum_{i=0}^{k-1} \left\| \omega_k^i f(x_i) - \omega_k^i m_k(x_i; B) \right\|_2^2 + \left\| B\Gamma - B_k^0 \Gamma \right\|_F^2 \end{aligned} \tag{4.35}$$

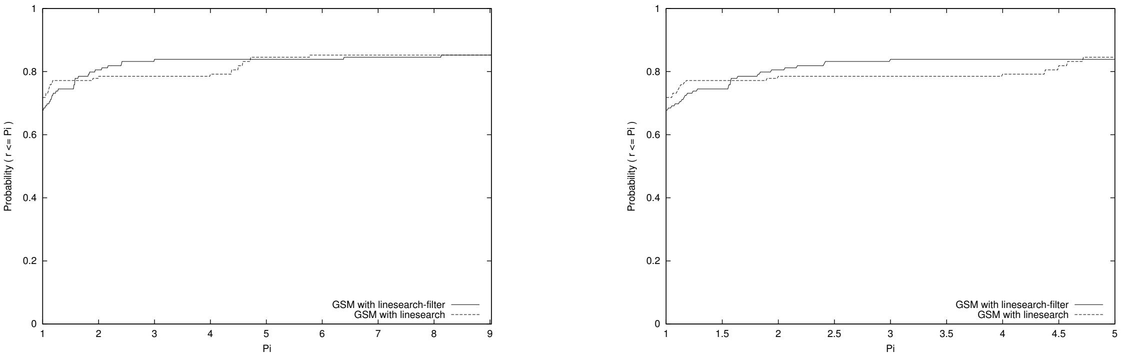


(a) Full profile

(b) Zoom

Figure 4.13: Performance profile with H0=I – GSM with linesearch-filter and linesearch –

At the moment, it is rather unclear how we could reformulate this least-squares to get a linear least-squares in $B$ as before (if we develop the above least-squares, we have $s_i^T B s_i$ terms which are problematic to treat). This would allow us to still be able to derive an update formula for $B_k$. If we cannot get a linear formulation, we should think about a way to efficiently solve this new least-squares problem. Consequently, numerical tests could be performed on singular optimization problems to compare the robustness and the efficiency of this approach with classical secant methods as well as our algorithms presented in Chapter 2.



(a) Full profile

(b) Zoom

Figure 4.14: Performance profile with H0=true Hessian – GSM with linesearch-filter and linesearch –

# Chapter 5

# Swiss behavioral models

## Contents

## 5.1 Introduction

We present behavioral models designed to capture the response of drivers to real-time traffic information. In 2003, we have conducted a survey in Switzerland in order to collect both Revealed Preferences (RP) and Stated Preferences (SP) about choice decisions in terms of route and mode. The RP data contains socioeconomic characteristics of the individuals in our samples, their actual usage of ITS as well as their actual route and mode choice behavior. The SP data provide us with stated route and mode choices when drivers are faced with different hypothetical choice situations involving real-time information about the state of the network. First we present a Mixed Binary Logit model with panel data to analyze the drivers' decisions when traffic information is provided during their trip by means of Radio Data System (RDS) or variable message signs (VMS). This model is referred to *en-route choice* model. Second we present Nested Logit models capturing the behavior of drivers when they are aware of traffic conditions before their trip. The latter models allow to predict *pre-trip route choice* decisions with regard to route and mode when traffic information is available. The calibrated models are subsequently included in a simulator which predicts travelers' behavior in specific scenarios (described by adjustable parameters) allowing the sensitivity analysis of the demand with regard to the variations of various parameters. In this chapter, we discuss the results of the estimation process, including some comments about the Value of Travel Time Savings (VTTS) and present some scenarios developed with our simulator.

The methodology presented in this chapter consists for most parts of ideas given in Bierlaire et al. (2006), which has been published in *European Transport*. It is also inspired from a report for the Swiss Federal Office of Roads (see Bierlaire and Thémans, 2006b).

## 5.2 Motivation and literature review

Intelligent Transportation Systems (ITS) are aiming at the improvement of transportation systems through advanced information and control technologies. Namely, Dynamic Traffic Management Systems (DTMS) combine those technologies with the appropriate decision-aid tools.

Demand models play a central role in such systems. Indeed, the impact of ITS on travelers' behavior must be captured, understood and explicitly predicted. In this context, representing transportation demand through (possibly dynamic) origin-destination matrices is not sufficient. A disaggregate representation is necessary, where individuals are considered with their characteristics (trip purpose, available ITS equipment, etc.) and with their decisions in terms of route and mode choice.

Most recent methodologies for the evaluation and management of ITS are based on behavioral models, predicting the response of users to the ITS environment. Among them, we can cite the software systems developed at the Massachusetts Institute of Technology: MITSIM Laboratory (Ben-Akiva et al., 1997) for the evaluation of DTMS and DynaMIT (Ben-Akiva et al., 2001) for real-time traffic information and prediction. Other tools, like VISSIM or AIMSUM in Europe, and DYNASMART and TRANSIM in the US are also based on a disaggregate representation of the demand.

The use of such tools allows for an operational approach of telematics, which optimizes the impact of existing infrastructures, such as Variable Message Signs (VMS), RDS, etc. Disaggregate demand models also help to analyze the impact of longer term strategies such as road-pricing, congestion-pricing, diversion strategies, etc.

During the last decade, various behavioral models have been proposed in the literature to capture response to traffic information. Although various methodologies have been used, such as cluster analysis (Conquest et al., 1993) or Poisson regression (Khattak et al., 2003), most approaches are based on discrete choice models. Khattak et al. (1996) present multinomial logit models estimated on both revealed preferences and stated preferences data. Wardman et al. (1997) and Chatterjee et al. (2002) propose a multinomial logit model capturing the response to information provided by Variable Message Signs. Mahmassani and Liu (1999) propose a Multinomial Probit model. Srinivasan and Mahmassani (2003) estimate a mixture of logit models (logit kernel) using a sample of commuters in the same city. We also refer the reader to Zhao (1996) and Dia (2002) for similar approaches.

In this chapter, we also adopt a discrete choice approach and present behavioral models capturing the response of Swiss travelers to traffic information, designed to be used in a DTMS. Compared to most approaches in the literature, we extend the analysis to both radio information and information coming from VMS, and consider SP data from different samples. As a consequence, we had to segment the population and include various socio-economic characteristics in the model. Also, in contrast to the existing literature (except for Conquest et al., 1993), we do not focus only on route-switching decisions. We consider also pre-trip mode-switching decisions. Finally, we adopt state-of-the-art models, such as a mixture of logit model with agent effects, and nested logit models jointly estimated on multiple data sets.

The models presented here are the result of a research project conducted between 2002 and 2004. The research team was composed of two engineering consulting firms (Robert-Grandpierre et Rapp, SA, Lausanne, and Büro Widmer, Frauenfeld), IVT (Institute for Transport Planning and Systems), ETH Zürich, and the Operations Research Group ROSO, EPFL.

The data collection process is described in Section 5.3. The model for en-route behavior is presented in Section 5.4 while the models for pre-trip behavior are presented in Section 5.5. Before concluding in Section 5.7, we illustrate examples of how these models can be used in a simulator in Section 5.6.

## 5.3 Data collection

Data was collected in two phases. In the first phase, the respondents were asked to report in a diary up to five trips performed during one given day, their associated use of advanced information systems, and their socioeconomic characteristics. The usual set of diary question was expanded to include items about the use of information systems, trip planning, time constraints, the route taken and alternative routes. It was clearly more difficult for the respondents than the usual diary. The revealed preference (RP) questionnaire included a question about the respondent's willingness to participate in the second phase of the study, involving a stated preferences (SP) experiment based on the answers in the RP diary. Each phase was separately pre-tested for response behavior and question quality. The surveys were undertaken in the spring (pre-test RP), summer (main study RP) and autumn (pre-test and main study SP) of 2003.

Three groups were targeted:

- commuters and car drivers in the French speaking canton Vaud. The addresses were provided by SIEMENS and the automobile club, TCS, which sent our diaries and reminders;
- commuters and car drivers in the German speaking canton Zürich. The addresses were provided by the automobile club, TCS, which sent our diaries and reminders;
- owners of a second home in Ticino from the German speaking part of the country, as they are very likely to undertake long-distance leisure journeys. The diary was adjusted to ask about the last relevant journey. The sample was constructed from public records about the owners of second homes in this canton south of the Alps.

The last group was designed to obtain long trips (typically, Zürich-Lugano represents 215km), as the impact of travel information is believed to be more significant for long distance trips.

The response to the RP survey is summarized in Table 5.1. A questionnaire was not considered useful if the description of the trips was not detailed enough, or if the longest reported trip was shorter than 7 km, a distance deemed necessary for information systems to have an impact on drivers' behavior. The value 7 km has been chosen to keep most inter-city trips in the sample.

| Response | Vaud | | Zürich | | Ticino | | Total | |
|---|---|---|---|---|---|---|---|---|
| Total sent | 826 | | 600 | | 323 | | 1749 | |
| Total received | 232 | | 195 | | 147 | | 574 | |
| Without reminder | | 180 | | 110 | | 62 | | 352 |
| After reminder | | 52 | | 85 | | 85 | | 222 |
| Usable | 223 | | 182 | | 137 | | 542 | |
| Share of usable | | | | | | | | |
| Responses [%] | 27 | | 30 | | 42 | | 31 | |

Table 5.1: Pre-test and main RP surveys: Response behavior

The response rates are low, both because only one reminder was possible and because of the complexity of the diary. The contrast between the travelers to the Ticino, for whom a congested journey is a regular occurrence and who already benefit from radio-distributed information, and the rest of the sample is striking. The increased response indicates an increased interest. The TCS based sample includes persons not working, as well as those never faced with congestion in the more rural parts of the respective cantons. Given that the changes between pre-test and main study were minor we included the usable responses from the pre-tests for the further analysis.

The stated preferences experiments were generated based on the longest reported trip (referred to as the "reference trip" in the rest of the chapter) of each respondent. The orthogonal experimental design generated by SPSS had been cleaned, so that no dominated choices remained. Each respondent received seven hypothetical pre-trip choice situations (route and mode choice) and seven hypothetical en-route choice situations (route choice only). In the pre-trip case, we assume that traffic information is available two hours before the trip starts. Three alternatives were presented in each case: the base alternative, an alternative recommended by the information system and a realistic public transportation alternative derived from the official timetable. The attribute values of the base alternative are based on those of the reported trip, in order to create a realistic choice context. The attributes of the two other alternatives were based on an orthogonal experimental design corrected for dominant alternatives.

The attributes for the road-based alternatives are

- Departure time,
- Estimated non-congested travel time
- Estimated congested travel time
- Estimated total travel time (the sum of the previous two)

- Percentage of error for the predicted times,
- Arrival time,
- Cost (operating costs including fuel, oil and maintenance).

Note that the percentage of error for the predicted times is meant to capture the overall perceived reliability of the information system.

The attributes of the public transportation alternative are

- Departure time from the closest public transportation stop.
- Travel time to the final stop (closest to the destination)
- Arrival time at the final stop (the sum of the two previous)
- Fare (accounting for yearly passes and specific discounts)

We excluded the public transport access and egress time to reduce the complexity of the presentation and because it is generally fixed and not under control of the service operator.

Having described alternatives in the pre-trip context, an hypothetical situation is obtained by giving realistic numerical values to the different characteristics of the above alternatives.

These values are calculated based on information about the reference trip which has been described by respondents in the RP phase of the survey.

**Desired arrival time** obtained by taking arrival time described for the reference trip and subtracting the possible minutes of delay or adding the possible minutes of early arrival.

**Free-flow travel time for the reference trip** calculated by using the software package "Route 66 2003 pour l'Autriche et la Suisse" allowing for door-to-door planning of itineraries. Note that we provided to the software the departure point, the destination as well as intermediate points described in the RP questionnaire.

**Distance for the reference trip** provided by the software mentioned above once the itinerary has been calculated.

**Car cost per kilometer** taking into account fuel consumption, oil consumption, and maintenance costs with regard to the car used in the reference trip.

**Departure time, departure station and stop station by public transportation** On the basis of the departure point and the destination for the reference trip, we have used the CFF website ( Swiss railways company `www.sbb.ch/en`) which allows for door-to-door planning in order to determine the best alternative by public transports. The arrival time at the end station was chosen such that it would allow to reach the destination at the desired time, accounting for the walking time between the end station and the final destination. The departure time and travel time for the public transportation alternatives were directly derived from this information.

**Cost by public transportation** The price of the train ticket was obtained from CFF website, taking into account possible discounts available to each respondent. For the rest of the trip (bus, subway, ...), we have used an experimental formula which is classical in such studies in Switzerland:

$$2.5 \log(\min(1, \text{length of the remaining of the trip}))$$

The numerical values used to describe alternatives of the pre-trip choice context have been obtained by using the factors contained in Table 5.2.

The columns of this table are labeled as follows:

**NBR** is the identifier of a set of factors.

**CF1** represents the congested travel time on route 1 and it is expressed in minutes.

**ERROR1** represents the error on information predicted for route 1 and it is expressed in percentage.

**FF2** represents the additional free-flow (non-congested) travel time for route 2 and it is expressed in minutes.

**CF2** represents the congested travel time for route 2 and it is expressed in minutes.

**ERROR2** represents the error on information predicted for route 2 and it is expressed in percentage.

**COST2** represents the multiplying factor for the cost of the trip on route 2 and it is expressed in percentage.

**PTT** represents the multiplying factor for the travel time by public transportation and it is expressed in percentage.

**TRADEOFF** tells us if the set of factors gives rise to a choice situation involving a trade-off or not: 1 if the choice requires a trade-off, 0 otherwise.

| NBR | CF1 | ERROR1 | FF2 | CF2 | ERROR2 | COST2 | PTTT | TRADEOFF |
|---|---|---|---|---|---|---|---|---|
| **1** | 10 | 5 | 18 | 10 | 8 | 110 | 85 | 0 |
| **7** | 10 | 5 | 18 | 5 | 12 | 90 | 85 | 1 |
| **29** | 15 | 2 | 18 | 10 | 3 | 110 | 85 | 0 |
| **27** | 10 | 2 | 8.5 | 10 | 8 | 110 | 100 | 0 |
| **8** | 15 | 2 | 18 | 5 | 3 | 90 | 85 | 1 |
| **13** | 10 | 2 | 8.5 | 5 | 12 | 90 | 90 | 1 |
| **19** | 15 | 5 | 8.5 | 10 | 3 | 110 | 90 | 1 |
| **21** | 15 | 5 | 8.5 | 5 | 3 | 90 | 100 | 1 |
| **3** | 10 | 2 | 4 | 0 | 3 | 90 | 85 | 1 |
| **10** | 10 | 5 | 4 | 0 | 3 | 90 | 90 | 0 |
| **17** | 10 | 2 | 4 | 0 | 3 | 110 | 85 | 1 |
| **18** | 10 | 5 | 4 | 0 | 3 | 110 | 100 | 1 |
| **31** | 25 | 2 | 18 | 0 | 3 | 90 | 100 | 1 |
| **15** | 25 | 2 | 18 | 0 | 3 | 110 | 90 | 1 |
| **28** | 15 | 2 | 4 | 0 | 8 | 90 | 100 | 1 |
| **32** | 15 | 5 | 4 | 0 | 8 | 90 | 85 | 1 |
| **6** | 25 | 2 | 4 | 10 | 12 | 90 | 85 | 1 |
| **2** | 25 | 5 | 4 | 10 | 12 | 90 | 100 | 1 |
| **16** | 15 | 2 | 4 | 0 | 12 | 110 | 90 | 1 |
| **20** | 15 | 5 | 4 | 0 | 12 | 110 | 85 | 1 |
| **4** | 25 | 2 | 4 | 5 | 8 | 110 | 85 | 1 |
| **24** | 25 | 5 | 4 | 5 | 8 | 110 | 90 | 1 |
| **26** | 25 | 5 | 8.5 | 0 | 3 | 90 | 85 | 0 |
| **23** | 25 | 5 | 8.5 | 0 | 3 | 110 | 85 | 1 |
| **25** | 45 | 5 | 18 | 0 | 8 | 90 | 90 | 0 |
| **22** | 45 | 5 | 18 | 0 | 12 | 110 | 100 | 1 |
| **9** | 45 | 2 | 4 | 10 | 3 | 90 | 90 | 0 |
| **5** | 45 | 5 | 4 | 10 | 3 | 90 | 85 | 0 |
| **14** | 45 | 2 | 4 | 5 | 3 | 110 | 100 | 1 |
| **11** | 45 | 5 | 4 | 5 | 3 | 110 | 85 | 1 |
| **12** | 45 | 2 | 8.5 | 0 | 8 | 90 | 85 | 0 |
| **30** | 45 | 2 | 8.5 | 0 | 12 | 110 | 85 | 1 |

Table 5.2: Factors for pre-trip experimental design

Among the 32 possible sets of factors in Table 5.2, we have kept only 23 sets presenting a trade-off. For each respondent, we chose randomly 7 sets of factors.

We present the way these values were actually computed. In Tables 5.3, 5.4 and 5.5, the column on the left contains the attributes of the alternative and the column on the right describes how they were computed. Information in *italic* corresponds to information calculated on the basis of the reference trip and information in **bold** comes from Table 5.2.

| **Route 1** | |
|---|---|
| Departure time | *Desired arrival time*<br>- estimated total travel time |
| Estimated non-congested travel time | *Free-flow travel time*<br>*for the reference trip* |
| Estimated congested travel time | **CF1** |
| Estimated total travel time | Sum of the previous two |
| Predicted arrival time | *Desired arrival time* |
| Error on predictions | **ERROR1** |
| Cost | *Distance for the reference trip*<br>× *Car cost per kilometer* |

Table 5.3: Computation of attributes for route 1

In the en-route case, we assume that traffic information is available during the trip. We also suppose that the radio is turned on and that there are VMS along the route. Two alternatives are included: the base alternative and alternative recommended by the information system. Their attributes are

- Estimated travel time to the destination from the current location
- Percentage of error on the predicted time

- Type of road to the destination: motorway and similar (labeled *national*), other roads (labeled *non-national*), or both,
- Source of information: Radio or Variable Message Signs (VMS)

The numerical values associated with the attributes described above are chosen in the Table 5.6.

The information contained in this table is:

**NBR** is the identifier of the set of factors.

**TT1** represents the remaining travel time on route 1 and it is expressed in minutes.

**ERROR1** represents the error on predictions for route 1 and it is expressed in percentage.

**MIX1** gives the type of road to the destination on route 1 using the following coding: 0 for *national* roads, 1 for Mix of *national* and *non-national* roads, and 2 for *non-national* roads.

**SOURCE1** gives the source of information on route 1 using the following coding: 1 for Radio and 2 for VMS.

**TT2** represents the remaining travel time on route 2 and it is expressed in minutes.

**ERROR2** represents the error on predictions for route 2 and it is expressed in percentage.

**MIX2** gives the type of road to the destination on route 2 using same coding as MIX1.

**SOURCE2** gives the source of information on route 2 using the same coding as SOURCE1.

**TRADEOFF** tells us if the set of factors gives rise to a choice situation involving a trade-off or not: 1 if the choice requires a trade-off, 0.5 if there is no trade-off and it is not straightforward to identify it, and 0 if there is obviously no trade-off.

Among the 27 possible sets of factors in Table 5.6, we have kept only 20 sets presenting a trade-off. For each respondent, we chose randomly 7 sets of factors.

The response to the SP survey is summarized in Table 5.7. A further 21 usable SP returns were obtained from the participants of the RP pre-test.

The response is a satisfactory 69%, which is normal after respondents have committed themselves to further participation. Table 5.14 compares the samples' characteristics with the Mikrozensus 2000, the national travel survey (Bundesamt für Raumentwicklung and Bundesamt für Statistik, 2001) for the usable 542 responses from the RP, and for the 186

SP questionnaires actually used in the pre-trip model. The shift in the sample structure is noticeable. While this shift is not a problem for parameter estimation[1], it is worth keeping it in mind. It reminds us, just how difficult SP experiments are and that SP designers should find new ways to present and construct the experiments. It also needs to be kept in mind during application, as any result will then need to be reweighted to the population means.

## 5.4 En-route model

A mixed logit model (see Train, 2003) for panel data has been estimated using the software package Biogeme (Bierlaire, 2003, Bierlaire, 2005). The specification of the two linear-in-parameters utility functions is reported in Table 5.8, where "radio" is 1 if information is received by the radio, 0 otherwise; "VMS" is 1 if information is received by VMS, 0 otherwise; "non-national" is 1 if the trip to the destination is using non-national roads, 0 otherwise; "frequent_usage" is 1 if the traveler frequently uses the radio to get traffic information, 0 otherwise; "unfrequent_usage" is "1-frequent_usage", that is 1 if the traveler does not frequently use the radio to get traffic information, 0 otherwise. The probability for individual $n$ of choosing alternative $i$ is given by

$$P_n(i|\{i,j\}) = \int_{\xi_n} \prod_t \frac{e^{V_{int}+\sigma_{\text{panel}}\xi_n}}{e^{V_{int}+\sigma_{\text{panel}}\xi_n} + e^{V_{jnt}}} f(\xi_n)d\xi_n$$

where the product ranges over all experiments $t$ of individual $n$, $\sigma_{\text{panel}}$ is an unknown parameter to be estimated, and $\xi_n$ is a standardized normal random parameter $\xi_n \sim N(0,1)$, so that

$$f(\xi_n) = \frac{1}{\sqrt{2\pi}} e^{-\xi_n{}^2/2},$$

and $V_{int}$ the utility associated by individual $n$ to alternative $i$ during experiment $t$. Note that the term $\sigma_{\text{panel}}\xi_n$ captures unobserved agent effects, constant over experiments.

A total of 1358 observations have been used (7 questions per respondent, 194 respondents). The estimated parameters are reported in Table 5.9.

All parameters are significant. We briefly discuss each of them.

$\beta_{\text{current}}$ is the Alternative Specific Constant associated with the first alternative. It is positive as expected. This captures a type of inertia to change.

$\beta_{\text{time}}$ is negative, as expected.

[1] Exogenous Sampling Maximum Likelihood provides consistent estimates for all parameters, see Manski and Lerman (1977), Manski and McFadden (1981) and Ben-Akiva and Lerman (1985, chap. 8)

$\beta_{\text{error\_radio\_freq}}$, $\beta_{\text{error\_radio\_unfreq}}$, $\beta_{\text{error\_vms}}$ are all negative, capturing the impact of uncertainty on travelers' choice, as people do not favor alternatives for which imprecise information is available. Comparing the three values, it appears that a same level of error is more penalized for a VMS than for the radio. Also, travelers who currently listen and use traffic information from the radio have a tendency to penalize the errors made by this media less. This could be explained by the fact that travelers have a better experience of radio than VMS.

$\beta_{\text{non-national}}$ is negative, capturing the fact that travelers are reluctant to leave the main road network. However, its absolute value is less than $\beta_{\text{current}}$, showing that, everything else being equal, travelers prefer their current route on non-national roads, rather than an alternative itinerary using national roads.

$\sigma_{\text{panel}}$ is significant, showing that it was important to include intra-personal effects in the model. Its sign is irrelevant.

Note that we have tried to estimate separate models for each subsample, but they did not appear to be significantly different.

## 5.5 Pre-trip models

We have estimated a joint nested logit model, combining a model for the Ticino sample (second home owners) and the rest of the sample (we did not discover any significant difference between the French and German speaking parts). The nested logit model is given by

$$P(i) = P(i|m)P(m) = \frac{e^{\mu_m V_i}}{\sum\limits_{j \in C_m} e^{\mu_m V_j}} \frac{e^{\mu \tilde{V}_m}}{\sum\limits_{k \in C} e^{\mu \tilde{V}_k}}$$

with

$$\tilde{V}_m = \frac{1}{\mu_m} \ln \sum_{i \in C_m} e^{\mu_m V_{im}}$$

where $i$ is one of the alternatives in the choice set $C = \{$`Route 1`, `Route 2`, `Public transportation`$\}$, $m$ is the nest containing $i$, that is either `Nest A` or `Nest B`, and $C_m$ is the set of alternatives within nest $m$. Tables 5.10 and 5.12 reports the linear-in-parameter specification of $V_i$.

The nested logit is a natural modeling approach to capture the correlation between the two car alternatives. Note that a mixed version of this model was also estimated to capture the unobserved agent effect. It appeared that it was not useful for the pre-trip models, as individual characteristics are already captured by fixed coefficients.

A total of 1302 observations have been used (7 questions per respondent, 186 respondents). A total of 34 parameters have been estimated: 2 nest parameters, one scale parameter, 11 parameters specific to the Ticino model, 16 specific parameters to the other model, and 4 parameters common to both models: $\beta_{\text{cost}}$, $\beta_{\text{error}}$, $\beta_{\text{radio\_usage}}$ and $\beta_{\text{profession}}$. The joint estimation appeared to be very useful to obtain efficient estimates of the common parameters.

- Initial log-likelihood: $\mathcal{L}(0) =$ -1399.63
- Final log-likelihood: $\mathcal{L}(\beta^*) =$ -767.245
- Rho-square: $\rho^2 = 0.451824$

Although jointly estimated, we present the results separately.

The specification of the Ticino model is reported in Table 5.10, where "frequent_usage" is 1 if the traveler frequently uses traffic information, 0 otherwise; "aware" is 1 if the traveler was informed by radio about the traffic state during the reference trip, 0 otherwise; "impact" is 1 if the traveler has actually used traffic information during the reference trip, 0 otherwise; "half-fare ticket" is 1 if the traveler owns a ticket which entitles to a 50% rebate on all main line services, 0 otherwise; "people" is the number of persons within the traveler's household; "cars" is the number of cars in the household; "manager" is 1 if the traveler is working as a manager or working at home, 0 otherwise; "income(>8'000 CHF)" is 1 if the monthly household income is above 8'000 CHF[2], 0 otherwise; "usage_percentage" is the percentage of public transportation trips among all trips to the second home.

Note that there is not enough variability in travel time and cost for the public transportation alternative in the Ticino sample, explaining why these attributes are not included in the model.

The results of the estimation are reported in Table 5.11. All parameters are significant at the 95% level of confidence, except $\beta_{\text{aware-Ticino}}$. However, the t-test is close to the 1.96 threshold. Therefore, we have decided to keep the parameter in the model.

$\beta_{\text{cost}}$ is negative, as expected for a travel cost coefficient.

$\beta_{\text{error}}$ is negative, as expected. Same conclusion as in the en-route model.

$\beta_{\text{radio\_usage}}$ is positive. It seems to show that the inertia is larger for frequent users of the traffic information at the radio. It is not clear if it is a feature of the model, or if the frequent usage of the radio indeed encourages inertia, because of bad experiences. This requires more investigation.

[2] In 2006, 1 CHF $\approx$ 0.645 €

$\beta_{\text{profession}}$ is negative, illustrating the aversion of managers and home-working persons to use public transportation.

$\beta_{\text{ASC1-Ticino}}$ and $\beta_{\text{ASC2-Ticino}}$ are the Alternative Specific Constants. There are positive, illustrating the attractiveness of the car versus public transportation.

$\beta_{\text{half\_fare-Ticino}}$ is positive, showing a propensity to use public transportation by the owners of a half-fare ticket.

$\beta_{\text{income-Ticino}}$ is positive, indicating the higher willingness of higher income travelers to shift, as they are better able to afford the costs of rail travel and of taxi as well as of related services after their journey. It is an indirect indicator of their higher value of time.

$\beta_{\text{aware-Ticino}}$ is negative, capturing an inertia, a preference toward the current alternative for more informed people. This is consistent with the comments about $\beta_{\text{radio\_usage}}$ (note that $\beta_{\text{aware-Ticino}}$ is in the utility function of the alternative route).

$\beta_{\text{impact-Ticino}}$ is positive, showing that people who have used traffic information to modify their decision during the reference trip have a propensity to change. It seems to support the assumption about the bad experience proposed in the analysis of the sign of $\beta_{\text{radio\_usage}}$.

$\beta_{\text{people\_nbr-Ticino}}$ is negative. Indeed, the marginal cost of one more person in the family is much more important for public transportation than for private transportation.

$\beta_{\text{car\_nbr-Ticino}}$ is negative. Indeed, the more cars in the household, the less likely the use of public transportation.

$\beta_{\text{public\_transportation-Ticino}}$ is positive, showing an attractivity for the public transportation by the most frequent users of public transportation.

$\beta_{\text{time\_jam1\_Ticino}}$ and $\beta_{\text{time\_jam2\_Ticino}}$ are both negative. The sensitivity to the predicted time in jam for the alternative route is more important. Note also that the free flow travel time did not appear significant in the model. It is due to the very low variability of this attribute for the Ticino sample.

The specification of the commuters model is reported in Table 5.12, where "d(0-50)" is 1 if the trip length is between 0 and 50km, 0 otherwise; "d(50-100)" is 1 if the trip length is between 50 and 100km, 0 otherwise; "frequent_usage" is 1 if the traveler frequently uses traffic information, 0 otherwise; "aware" is 1 if the traveler was informed by radio about the traffic state during the reference trip, 0 otherwise; "manager" is 1 if the traveler is working as a manager or working at home, 0 otherwise; "early_arrival" is the number of minutes between the arrival by public transportation and the scheduled arrival time; "fare" is the public transportation fare; "timetable" is the scheduled travel time from the timetable; "age(0-40)" is 1 if the traveler is younger than 40, 0 otherwise; "car_as_mode" is 1 if the car was the chosen mode for the reference trip, 0 otherwise; "car_availability" is 1 if a car is available to the traveler, 0 otherwise[3]; "car_type" is 1 if a company car has been used during the reference trip, 0 otherwise; "kilometers" is the number of kilometers traveled by car per year.

The results of the estimation are reported in Table 5.13. All parameters are significant to the 95% level of confidence, except $\beta_{\text{internet\_usage}}$ and $\beta_{\text{fare}}$. However, the t-tests are close to the 1.96 threshold value, and we have decided to keep them in the model.

Parameters $\beta_{\text{cost}}$, $\beta_{\text{error}}$, $\beta_{\text{radio\_usage}}$ and $\beta_{\text{profession}}$ have been discussed above.

$\beta_{\text{ASC1}}$ and $\beta_{\text{ASC2}}$ are the Alternative Specific Constants for the two first alternatives. They are negative, which is difficult to interpret. Indeed, the cost and time parameters are alternative specific. For instance, if we compare alternatives with a cost of 10 CHF, a travel time of 50 minutes (both for car and public transportation), the probability of choosing the public transportation is significantly smaller than the probability to choose the car, as expected.

$\beta_{\text{mode}}$ is negative, meaning that people reporting to use their car have a preference toward the car, so it affects negatively the public transportation alternative.

$\beta_{\text{availability}}$ is negative, meaning that people who have a car available have a tendency to use it, so it affects negatively the public transportation alternative.

$\beta_{\text{type}}$ is negative, for the same reason as described above.

$\beta_{\text{internet\_usage}}$ is negative, showing that people who use Internet to access the information have a propensity to switch route. It is interesting to note that the parameter $\beta_{\text{radio\_usage}}$ is positive in comparison.

$\beta_{\text{aware}}$ is positive, showing that people who are aware of alternative routes, have a propensity to switch. Note that, in comparison to the Ticino model, the commuter model deals with situations where the number of feasible routes is usually higher.

$\beta_{\text{age}}$ is negative, showing that people younger than 40 have a preference for the car.

[3] Car availability is understood by respondents as a question about car ownership. Other cars can still be available to license holders, such as those from the popular car-sharing firm "Mobility" or those of family and friends.

$\beta_{\text{kms}}$ is negative, showing that the more the car is used per year, the less appealing public transportations are.

$\beta_{\text{early}}$ is negative, capturing the inconvenience of mismatch between the actual arrival time and desired arrival time when using public transportation.

$\beta_{\text{fare}}$ is negative, as expected for a cost coefficient. Note that it is less negative than the cost coefficient for the car alternatives.

$\beta_{\text{timetable}}$ is negative, as expected for a travel time coefficient.

$\beta_{\text{time\_jam\_medium}}$, $\beta_{\text{time\_jam\_short}}$, $\beta_{\text{time\_free\_medium}}$, $\beta_{\text{time\_free\_short}}$ are all negative, as expected. As discussed below, although they have the correct sign, we are somehow suspicious about the parameters estimates for the short trips. Indeed, there are plenty of context-specific constraints associated with short trips that are not accounted for in this model. The fact that travel time in free flow conditions is more penalized than travel time in jam is counter-intuitive. In the "medium" case (trips between 50 and 100km), travel time in traffic jam is more penalized than travel time in free flow conditions.

It is interesting to analyze the Value Travel Time Savings (VTTS), as provided by the commuter model. As we use a linear specification, this quantity is simply given by the ratio between the travel time coefficient and the travel cost coefficient.

| **VTTS (CHF/min)** | Free flow | in Jam |
|---:|:---:|:---:|
| Short distance ($\leq$ 50km) | *50.7* | 34.8 |
| Medium distance ($>$ 50km) | 27.3 | 36.5 |

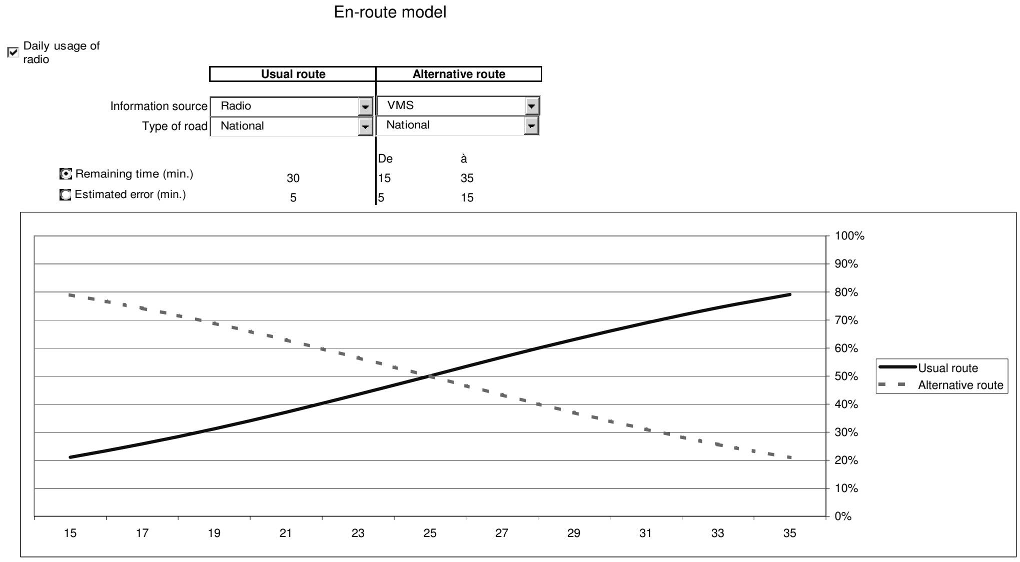
The values for the medium distances are comparable with the results provided by Koenig et al. (2004): 35.9 CHF, assuming an income of 10'000 CHF/month and a business trip of 75km. However, for the short distance, our values are significantly higher. Koenig et al. (2004) obtain 24.22 CHF, assuming an income of 10'000 CHF/month and a business trip of 25km. Clearly, in our model, we have a low granularity of distances and travel times for short distance trips. The approach by Koenig et al. (2004) is more appropriate to estimate VTTS for short trips. Anyway, the value 50.7 CHF, reported in italic above, does not seem valid to us. We believe the time and cost parameters capture other effects associated with short trips, that should be explicitly analyzed.

Note that it appeared that adding an error component to capture the agent effect was not useful for the pre-trip models, as individual characteristics are already captured by fixed coefficients.

## 5.6 Simulation

The models presented above are based on stated preference data. Like any such models, they cannot directly be used for the prediction of market shares, but are very useful for policy analysis using "what-if" scenarios. We have therefore implemented a simulator based on the estimated models. The simulator is an Excel sheet available from the authors upon request. We have selected here a couple of illustrative examples based on the en-route model, to give a flavor of the results.

Figure 5.1 is a screen-shot of the simulator for the En-route model, where the probability of the two alternatives is presented as a function of the predicted travel time on the alternative route, ranging from 15 to 35 minutes. In this scenario, the predicted travel time on the usual route is assumed to be 30 minutes, the error on the information is 5 minutes for both alternatives, the source of information is radio for the usual route and VMS for the alternative route, and the individual is assumed to have a daily usage of the radio. The type of road is "national" for both alternatives. Among other things, it is interesting to note that the 50% probability is reached when the alternative route is 25 minutes, compared to the 30 minutes on the usual route. Also, if both routes are said to be 30 minutes, the probability to switch route is only about 34%, illustrating the inertia to change.



En-route model

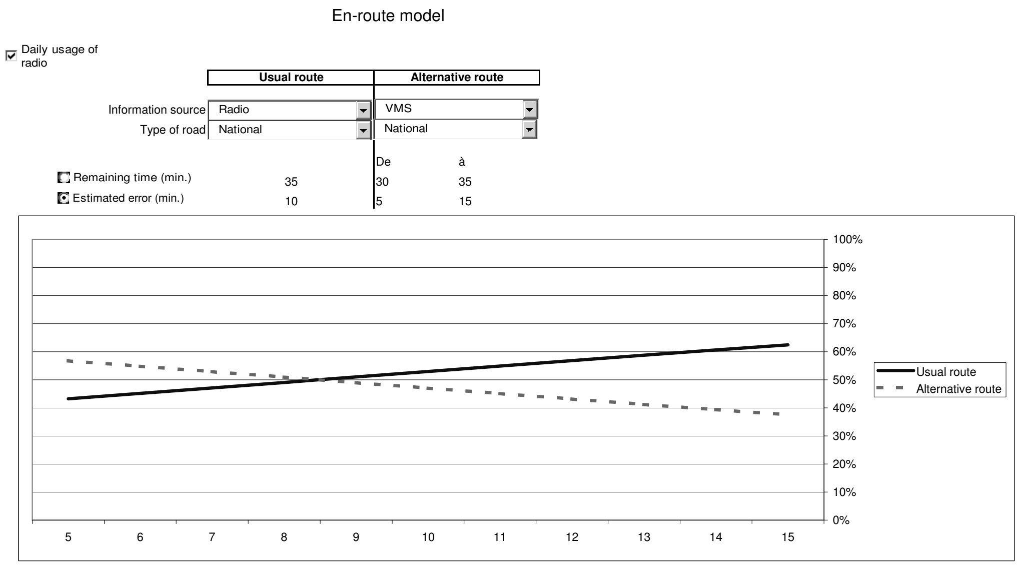
☑ Daily usage of radio

| | Usual route | Alternative route | |
|---|---|---|---|
| Information source | Radio | VMS | |
| Type of road | National | National | |
| | | De | à |
| ● Remaining time (min.) | 30 | 15 | 35 |
| ○ Estimated error (min.) | 5 | 5 | 15 |

Figure 5.1: First scenario

Figure 5.2 is also a screen-shot of the simulator for the En-route model, where the probability of the two alternatives is presented as a function of the estimated error on the alternative route, ranging from 5 to 15 minutes. In this scenario, the error on the information about the usual route is assumed to be 10 minutes, the predicted travel time is 35 minutes on the usual route and 30 minutes on the alternative route, the source of information is radio for the usual route and VMS for the alternative route, and the individual is assumed to have a daily usage of the radio. The type of road is “national” for both alternatives.

Note that 50% probability is reached for a value of about 8.5. If both errors are 10 minutes, the probability to switch is about 47%.



En-route model

☑ Daily usage of radio

| | Usual route | Alternative route |
|---|---|---|
| Information source | Radio | VMS |
| Type of road | National | National |

| | | De | à |
|---|---|---|---|
| ○ Remaining time (min.) | 35 | 30 | 35 |
| ● Estimated error (min.) | 10 | 5 | 15 |

Figure 5.2: Second scenario

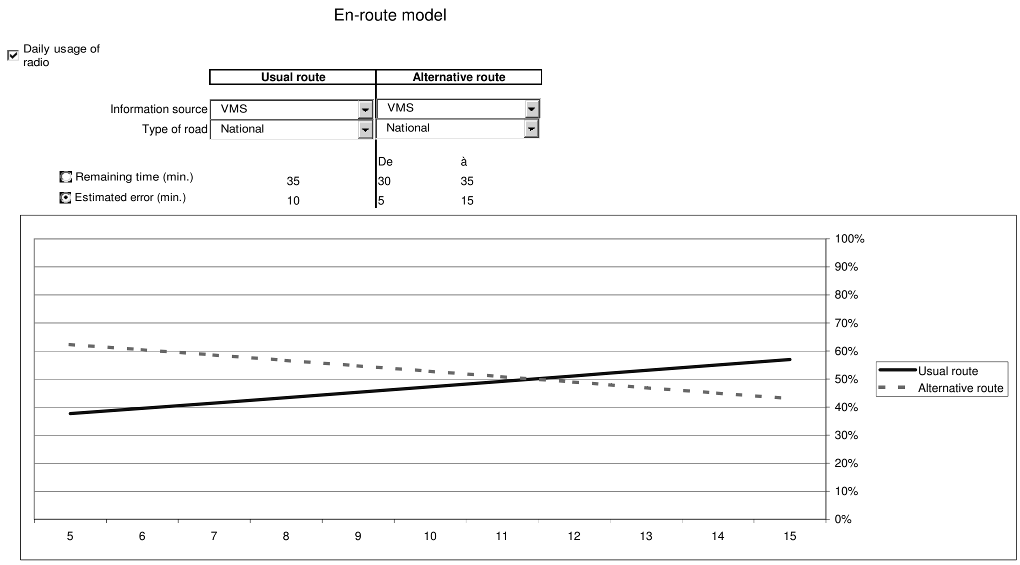
Finally, Figure 5.3 illustrates the same scenario as Figure 5.2, except that the information about the usual route is obtained from a VMS instead of the radio. We note that the 50% value shifts from about 8.5 to about 11.5, illustrating that travelers have less confidence in VMS, everything else being equal.

## 5.7 Conclusions and perspectives

We have estimated a model capturing the response to en-route information, and two models capturing the response to pre-trip information, based on data collected in Switzerland during 2003.

The en-route model enables to measure the level of inertia to en-route switching and the preference toward national roads, among other things. It has been illustrated using some examples of the simulator.

In the pre-trip models, the heterogeneity of the sample has been emphasized. Indeed, the socioeconomic characteristics play a significant role in these models. First, a model


En-route model

☑ Daily usage of radio

| | Usual route | Alternative route | |
|---|---|---|---|
| Information source | VMS | VMS | |
| Type of road | National | National | |
| | | De | à |
| Remaining time (min.) | 35 | 30 | 35 |
| Estimated error (min.) | 10 | 5 | 15 |

Figure 5.3: Third scenario

for the owners of a second home in Ticino has been estimated. It allows to capture and predict the important role of traffic information, and of public transportation in this specific context, and may help to design appropriate focused policies for long distance, non-work related, trips. Second, a model for commuters has been estimated. While the model seems valid for medium distance trips, we have significant suspicions of its validity for short distance trips. More investigation is necessary to better understand the constraints and the choice context of such trips. The attributes included in our SP experiments are probably not sufficient to explain them.

The models that have been estimated are advanced random utility models. The en-route model is a mixed binary logit model with panel data. The pre-trip models are heterogeneous nested logit models. They have all been estimated using the BIOGEME software package.

We conclude by mentioning some potentially interesting streams of investigations:

- The diversity of behaviors emphasized in this study suggests the development of regular surveys to better understand this phenomenon. The cost of collecting such data being important, organizing regular surveys would also bring very valuable information at a low marginal cost. Moreover, it would allow to analyze the behavioral dynamics, in order to understand how travelers change their behavior as they experience the use of ITS.

- The abnormally high VTTS for short distance trips should be investigated. For instance, mixed GEV models could be considered, along the lines discussed by Hess et al. (2005b).

- It appears from the models that the level of error in an information system significantly influences its perception. However, this concept has been kept at an abstract level in our surveys, and would deserve a deeper analysis.

- Our sample is biased toward private car users. A more systematic analysis of mode choice would require more public transportation users in the sample.

The use of demand models is more and more critical in the ITS context. The models estimated in this chapter allows to better understand and predict the response of travelers to traffic information. From a system design point of view, the most notable conclusions of our study are linked to

- the willingness of the respondents to act when informed

- the impact of errors in the information

The willingness to act invites further investment into information provision, both en-route and pre-trip. It invites specifically investment in information with little error (see the relatively high trade-offs, which the respondents' parameters imply). This is a real challenge, as error-free information is based on both fast and reliable data collection, as well as on a system which can anticipate the response of the drivers to any information.

| **Route 2** | |
|---|---|
| Departure time | *Desired arrival time*<br>- estimated total travel time |
| Estimated non-congested travel time | *Free-flow travel time*<br>*for the reference trip* + **FF2** |
| Estimated congested travel time | **CF2** |
| Estimated total travel time | Sum of the previous two |
| Predicted arrival time | *Desired arrival time* |
| Error on predictions | **ERROR2** |
| Additional distance | **FF2** × 60 km/h |
| Cost | (*Distance for the reference trip*<br>+ additional distance)<br>× *Car cost per kilometer*<br>× (**COST2**/100) |

Table 5.4: Computation of attributes for route 2

| **Public transportation** | |
|---|---|
| Departure time | *Departure time by public transportation* |
| Estimated travel time | *Duration* × (**PTTT**/100) |
| Predicted arrival time | Departure time + estimated travel time |
| Cost | *Cost by public transportation* |

Table 5.5: Computation of attributes for public transportation

| NBR | TT1 | ERROR1 | MIX1 | SOURCE1 | TT2 | ERROR2 | MIX2 | SOURCE2 | TRADEOFF |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 25 | 10 | 2 | 2 | 25 | 10 | 1 | 2 | 1 |
| **2** | 30 | 2 | 2 | 1 | 15 | 15 | 0 | 1 | 1 |
| **3** | 45 | 2 | 1 | 2 | 15 | 10 | 0 | 2 | 0 |
| **4** | 25 | 2 | 1 | 1 | 35 | 5 | 1 | 2 | 1 |
| **5** | 45 | 10 | 1 | 2 | 15 | 15 | 2 | 1 | 1 |
| **6** | 25 | 10 | 1 | 1 | 35 | 10 | 0 | 1 | 1 |
| **7** | 30 | 2 | 0 | 2 | 35 | 15 | 1 | 1 | 1 |
| **8** | 25 | 10 | 0 | 1 | 15 | 10 | 2 | 1 | 1 |
| **9** | 30 | 10 | 1 | 1 | 25 | 5 | 1 | 1 | 0 |
| **10** | 45 | 5 | 2 | 1 | 35 | 5 | 2 | 1 | 0 |
| **11** | 25 | 5 | 2 | 2 | 25 | 15 | 0 | 1 | 1 |
| **12** | 30 | 10 | 0 | 2 | 35 | 5 | 0 | 1 | 1 |
| **13** | 45 | 10 | 2 | 1 | 35 | 15 | 0 | 2 | 1 |
| **14** | 30 | 5 | 0 | 2 | 35 | 10 | 2 | 2 | 1 |
| **15** | 30 | 2 | 1 | 1 | 25 | 15 | 2 | 2 | 1 |
| **16** | 45 | 2 | 0 | 1 | 25 | 10 | 2 | 1 | 1 |
| **17** | 30 | 5 | 2 | 1 | 15 | 10 | 1 | 1 | 0 |
| **18** | 30 | 5 | 1 | 1 | 25 | 10 | 0 | 1 | 0 |
| **19** | 25 | 5 | 0 | 1 | 15 | 15 | 1 | 2 | 1 |
| **20** | 45 | 5 | 0 | 1 | 25 | 5 | 0 | 2 | 0 |
| **21** | 25 | 2 | 2 | 2 | 25 | 5 | 2 | 1 | 1 |
| **22** | 45 | 2 | 2 | 1 | 35 | 10 | 1 | 1 | 1 |
| **23** | 45 | 5 | 1 | 2 | 15 | 5 | 1 | 1 | 0.5 |
| **24** | 30 | 10 | 2 | 1 | 15 | 5 | 2 | 2 | 0.5 |
| **25** | 25 | 5 | 1 | 1 | 35 | 15 | 2 | 1 | 0 |
| **26** | 25 | 2 | 0 | 1 | 15 | 5 | 0 | 1 | 1 |
| **27** | 45 | 10 | 0 | 1 | 25 | 15 | 1 | 1 | 1 |

Table 5.6: Factors for on-trip experimental design

| Response | Vaud | Zürich | Ticino | Total |
|---|---|---|---|---|
| Total sent | 103 | 91 | 86 | 280 |
| Total received | 71 | 65 | 72 | 208 |
| Without reminder | 52 | 31 | 36 | 119 |
| After 2 reminders | 19 | 34 | 36 | 89 |
| Usable (en-route model) | 65 | 63 | 66 | 194 |
| Usable (pre-trip model) | | | | 186 |
| Share of usable | | | | |
| Responses [%] | 63 | 69 | 77 | 69 |

Table 5.7: Main SP survey: Response behavior

| | Current | Alternative |
|---|---|---|
| | route | route |
| $\beta_{\text{current}}$ | 1 | 0 |
| $\beta_{\text{time}}$ | remaining time | remaining time |
| $\beta_{\text{error\_radio\_freq}}$ | error * radio * frequent_usage | error * radio * frequent_usage |
| $\beta_{\text{error\_radio\_unfreq}}$ | error * radio * unfrequent_usage | error * radio * unfrequent_usage |
| $\beta_{\text{error\_vms}}$ | error * VMS | error * VMS |
| $\beta_{\text{non-national}}$ | non-national | non-national |

Table 5.8: En-route model specification

| Name | Value | Std error | t-test |
|---|---|---|---|
| $\beta_{\text{current}}$ | 0.552 | 0.110 | 5.015 |
| $\beta_{\text{time}}$ | -0.133 | 0.012 | -10.869 |
| $\beta_{\text{error\_radio\_freq}}$ | -0.055 | 0.016 | -3.405 |
| $\beta_{\text{error\_radio\_unfreq}}$ | -0.076 | 0.023 | -3.352 |
| $\beta_{\text{error\_vms}}$ | -0.078 | 0.016 | -4.938 |
| $\beta_{\text{non-national}}$ | -0.270 | 0.101 | -2.679 |
| $\sigma_{\text{panel}}$ | -0.716 | 0.156 | -4.576 |

K= 7

$\mathcal{L}(0)=$ -940.601

$\mathcal{L}(\beta^*)=$ -701.949

$\rho^2 = 0.254$

$\bar{\rho}^2 = 0.246$

Table 5.9: Estimated parameters of the en-route model

| | Nest A | | Nest B |
|---|---|---|---|
| | Route 1 | Route 2 | Public transportation |
| $\beta_{\text{ASC1-Ticino}}$ | 1 | 0 | 0 |
| $\beta_{\text{ASC2-Ticino}}$ | 0 | 1 | 0 |
| $\beta_{\text{cost}}$ | cost | cost | - |
| $\beta_{\text{error}}$ | error | error | - |
| $\beta_{\text{time\_jam1-Ticino}}$ | time in jam | - | - |
| $\beta_{\text{time\_jam2-Ticino}}$ | - | time in jam | - |
| $\beta_{\text{radio\_usage}}$ | frequent_usage | - | - |
| $\beta_{\text{aware-Ticino}}$ | - | aware | - |
| $\beta_{\text{impact-Ticino}}$ | - | impact | - |
| $\beta_{\text{half\_fare-Ticino}}$ | - | - | half-fare ticket |
| $\beta_{\text{people\_nbr-Ticino}}$ | - | - | people |
| $\beta_{\text{car\_nbr-Ticino}}$ | - | - | cars |
| $\beta_{\text{profession}}$ | - | - | manager |
| $\beta_{\text{income-Ticino}}$ | - | - | income(>8000CHF) |
| $\beta_{\text{public\_transportation-Ticino}}$ | - | - | usage_percentage |

Table 5.10: Specification of the pre-trip model for Ticino

| Name | Value | Std error | t-test |
|---|---|---|---|
| $\beta_{\text{cost}}$ | -0.145 | 0.034 | -4.214 |
| $\beta_{\text{error}}$ | -0.021 | 0.009 | -2.209 |
| $\beta_{\text{radio\_usage}}$ | 0.401 | 0.125 | 3.218 |
| $\beta_{\text{profession}}$ | -2.297 | 0.409 | -5.613 |
| $\beta_{\text{ASC1-Ticino}}$ | 12.11 | 3.225 | 3.754 |
| $\beta_{\text{ASC2-Ticino}}$ | 12.67 | 3.293 | 3.847 |
| $\beta_{\text{half\_fare-Ticino}}$ | 2.386 | 0.862 | 2.768 |
| $\beta_{\text{income-Ticino}}$ | 3.186 | 1.314 | 2.425 |
| $\beta_{\text{aware-Ticino}}$ | -0.354 | 0.182 | *-1.942* |
| $\beta_{\text{impact-Ticino}}$ | 0.505 | 0.196 | 2.579 |
| $\beta_{\text{people\_nbr-Ticino}}$ | -1.210 | 0.391 | -3.094 |
| $\beta_{\text{car\_nbr-Ticino}}$ | -1.173 | 0.446 | -2.634 |
| $\beta_{\text{public\_transportation-Ticino}}$ | 0.190 | 0.053 | 3.579 |
| $\beta_{\text{time\_jam1\_Ticino}}$ | -0.048 | 0.014 | -3.322 |
| $\beta_{\text{time\_jam2\_Ticino}}$ | -0.073 | 0.025 | -2.967 |
| $\mu_{\text{Nest A-Ticino}}$ | 4.057 | 0.971 | $3.147^*$ |
| $\lambda_{\text{scale}}$ | 0.580 | 0.151 | $-2.787^*$ |

Superscript $^*$ means that the t-test is against 1

Table 5.11: Estimated parameters for the Ticino pre-trip model

| | Nest A | | Nest B |
|---|---|---|---|
| | Route 1 | Route 2 | Public transp. |
| $\beta_{\mathrm{ASC1}}$ | 1 | 0 | 0 |
| $\beta_{\mathrm{ASC2}}$ | 0 | 1 | 0 |
| $\beta_{\mathrm{cost}}$ | cost | cost | - |
| $\beta_{\mathrm{error}}$ | error | error | - |
| $\beta_{\mathrm{time\_jam-short}}$ | time in jam * d(0-50) | time in jam * d(0-50) | - |
| $\beta_{\mathrm{time\_jam-medium}}$ | time in jam * d(50-100) | time in jam * d(50-100) | - |
| $\beta_{\mathrm{time\_free-short}}$ | fr. flow time * d(0-50) | fr. flow time * d(0-50) | - |
| $\beta_{\mathrm{time\_free-medium}}$ | fr. flow time * d(50-100) | fr. flow time * d(50-100) | - |
| $\beta_{\mathrm{radio\_usage}}$ | frequent_usage | - | - |
| $\beta_{\mathrm{internet\_usage}}$ | frequent_usage | - | - |
| $\beta_{\mathrm{aware}}$ | - | aware | - |
| $\beta_{\mathrm{early}}$ | - | - | early arrival |
| $\beta_{\mathrm{fare}}$ | - | - | fare |
| $\beta_{\mathrm{timetable}}$ | - | - | timetable |
| $\beta_{\mathrm{profession}}$ | - | - | manager |
| $\beta_{\mathrm{age}}$ | - | - | age(0-40) |
| $\beta_{\mathrm{mode}}$ | - | - | car_as_mode |
| $\beta_{\mathrm{availability}}$ | - | - | car_availability |
| $\beta_{\mathrm{type}}$ | - | - | car_type |
| $\beta_{\mathrm{kms}}$ | - | - | kilometers |

Table 5.12: Specification of the pre-trip model for commuters

| Name | Value | Std error | t-test |
|---|---|---|---|
| $\beta_{\text{cost}}$ | -0.145 | 0.034 | -4.214 |
| $\beta_{\text{error}}$ | -0.021 | 0.009 | -2.209 |
| $\beta_{\text{radio\_usage}}$ | 0.401 | 0.125 | 3.218 |
| $\beta_{\text{profession}}$ | -2.297 | 0.409 | -5.613 |
| $\beta_{\text{ASC1}}$ | -3.054 | 1.144 | -2.670 |
| $\beta_{\text{ASC2}}$ | -2.780 | 1.141 | -2.436 |
| $\beta_{\text{mode}}$ | -1.390 | 0.297 | -4.683 |
| $\beta_{\text{availability}}$ | -3.659 | 1.081 | -3.386 |
| $\beta_{\text{type}}$ | -3.016 | 1.093 | -2.760 |
| $\beta_{\text{internet\_usage}}$ | -0.239 | 0.125 | *-1.910* |
| $\beta_{\text{aware}}$ | 0.708 | 0.156 | 4.523 |
| $\beta_{\text{age}}$ | -1.197 | 0.341 | -3.513 |
| $\beta_{\text{kms}}$ | -0.041 | 0.012 | -3.420 |
| $\beta_{\text{early}}$ | -0.033 | 0.011 | -3.166 |
| $\beta_{\text{fare}}$ | -0.037 | 0.022 | *-1.674* |
| $\beta_{\text{timetable}}$ | -0.066 | 0.009 | -7.019 |
| $\beta_{\text{time\_jam\_medium}}$ | -0.088 | 0.019 | -4.543 |
| $\beta_{\text{time\_jam\_short}}$ | -0.084 | 0.015 | -5.582 |
| $\beta_{\text{time\_free\_medium}}$ | -0.066 | 0.011 | -5.752 |
| $\beta_{\text{time\_free\_short}}$ | -0.122 | 0.015 | -8.081 |
| $\mu_{\text{Nest A}}$ | 1.951 | 0.311 | $3.051^*$ |
| $\lambda_{\text{scale}}$ | 0.580 | 0.151 | $-2.787^*$ |

Superscript $^*$ means that the t-test is against 1

Table 5.13: Estimated parameters for the pre-trip commuters model

| | Nat. Travel<br>survey 2000 | Usable RP | | SP used | |
|---|---|---|---|---|---|
| **Sex** | | | | | |
| Male | 46.4% | 354 | 65.3% | 122 | 65.6% |
| Female | 53.7% | 188 | 34.7% | 64 | 34.4% |
| **Education** | | | | | |
| Primary+lower secondary | 34.0% | 30 | 5.5% | 4 | 2.2% |
| Vocational training | 40.7% | 252 | 46.5% | 76 | 40.9% |
| A-level, tertiary | 25.3% | 260 | 48.0% | 106 | 57.0% |
| **Working status** | | | | | |
| None | 47.4% | 113 | 20.8% | 36 | 19.4% |
| Employed | 46.8% | 358 | 66.1% | 126 | 67.7% |
| Self-employed | 5.8% | 71 | 13.1% | 24 | 12.9% |
| **Driving license** | | | | | |
| Yes | 78.4% | 493 | 91.0% | 176 | 94.6% |
| No | 21.6% | 49 | 9.0% | 10 | 5.4% |
| **Railpass "General abonment"** | | | | | |
| Yes | 6.0% | 61 | 11.3% | 20 | 10.8% |
| No | 94.0% | 481 | 88.7% | 166 | 89.2% |
| **Half-fare card** | | | | | |
| Yes | 34.8% | 379 | 69.9% | 138 | 74.2% |
| No | 63.2% | 163 | 30.1% | 48 | 25.8% |
| **Income [CHF]** | | | | | |
| < 2K | 3.1% | 5 | 0.9% | 0 | 0.0% |
| 2K-4K | 14.8% | 34 | 6.3% | 8 | 4.3% |
| 4K-6K | 22.5% | 90 | 16.6% | 23 | 12.4% |
| 6K-8K | 16.2% | 125 | 23.1% | 46 | 24.7% |
| 8K-10K | 9.7% | 109 | 20.1% | 51 | 27.4% |
| 10K-12K | 5.2% | 51 | 9.4% | 21 | 11.3% |
| 12K-14K | 2.6% | 42 | 7.7% | 17 | 9.1% |
| > 14K | 4.0% | 45 | 8.3% | 17 | 9.1% |
| No response | 21.9% | 41 | 7.6% | 3 | 1.6% |

Table 5.14: Socioeconomic characteristics

# Chapter 6

# Conclusion

## Contents

## 6.1 Review of main results

This dissertation proposes several algorithmic advances in important domains of numerical optimization, namely:

- unconstrained nonlinear optimization
- systems of nonlinear equations

The developments we present are motivated by the following contexts of real application (see Chapter 1):

- maximum likelihood estimation of discrete choice models
- Dynamic Traffic Management Systems

Algorithmic methods proposed are designed to be efficient in solving optimization problems in such contexts. During the development of algorithms dedicated to real applications involving simulation tools, the main objective is that these algorithms have to be able to compute a solution of the optimization problem under consideration in a limited amount of time. In addition, it arises that a given budget of computational cost cannot be violated. We have developed the different optimization methods presented in this dissertation keeping in mind this goal.

We have first developed two trust-region based algorithms capable of dealing with singularities in the objective function of an unconstrained nonlinear optimization problem (see Chapter 2). The identification of a singularity in the objective function once the iterates reach a local minimum is achieved by performing a QR-like analysis of the second derivatives matrix with a generalization of a classical technique in numerical algebra, that is the inverse iteration. This information about the singularity is consequently used to artificially add curvature using a penalty approach. These modifications have been integrated in both trust-region and filter frameworks. Numerical results have shown that the proposed methods perform better in terms of efficiency and robustness than classical algorithms of the literature on a large set of singular problems. The computational overhead of the identification procedure is compensated by the important decrease in the number of function evaluations necessary to converge to a solution. This makes the method particularly appealing for problems where the CPU time spent in function evaluations is important, such as those involving simulation.

In Chapter 3, we have presented a new heuristic for nonlinear global optimization, which aims to identify a global minimum of general unconstrained nonlinear optimization problems. The proposed variable neighborhood search framework makes use of an efficient trust-region algorithm (approximating derivatives of the objective function up to order 2) as local search in order to quickly converge to minima of the problem. This VNS heuristic is also able to prematurely stop the local search if the iterates are converging to a local minimum which has already been visited or if they are reaching an area where no significant improvement can be expected. This feature allows to drastically limit the application of the local search, and consequently, the number of function evaluations. Information about the objective function and its derivatives, namely its curvature, is also used to define the list of neighborhoods as well as the neighbors selection procedure. This definition of neighborhoods prevents the algorithm from being stucked in valleys closed to minima already identified and allows to jump toward surrounding valleys. Numerical experiments have been conducted very successfully as the proposed heuristic is more robust and more efficient than classical competitors from the literature. Most importantly, this heuristic allows to significantly reduce the average number of function iterations necessary to identify a global minimum. This better efficiency brings also improvements in computational time. Although the development of this heuristic is motivated by the non-concavity of the log-likelihood function involved in discrete choice models estimation (see Chapter 1), its application is very general.

Chapter 4 is dedicated to the resolution of systems of nonlinear equations. The contribution in this field consists in a new secant method, called GSM, which is based on a population of previous iterates and which uses a least squares approach to construct the linear model of the system to be solved. This new approach leads to an update formula and can be viewed as a generalization of the well-known Broyden's method. In addition to this generalized secant method, we have proposed two globalization techniques designed for Newton-like methods in the context of systems of nonlinear equations. Numerical experiments compare GSM with classical quasi-Newton methods of the literature. The proposed globalization techniques highly improve the robustness of all challenged secant methods. The numerous tests performed on a large set of systems have showed evidence that GSM is more robust and more efficient than other secant methods, including Broyden's methods. GSM significantly reduces the number of system evaluations in order to get a solution on the majority of the test-problems. Globalized versions of GSM are shown to be competitive both in terms of robustness and efficiency with Newton-Krylov methods which are based on derivatives approximations using finite differences. Even if the development of the GSM method has been motivated by transportation applications in this dissertation (see Chapter 1), it can be potentially applied in many other contexts of application involving systems of nonlinear equations, possibly expensive to evaluate and requiring simulation. For instance, Crittin (2004) has applied GSM to solve problems of consistent anticipatory route guidance.

In Chapter 5, we have developed behavioral models using discrete choice analysis in order to predict transportation demand in Switzerland in the context of real-time applications providing traffic information. These models are able to capture the impact of real-time information about traffic conditions on drivers' decisions in terms of route as well as mode choices. On the one hand, the en-route model enables to measure the level of inertia of drivers to route switching and their preference for national roads and highways. We can also conclude that drivers are more confident in radio information than in information obtained from variable message signs. On the other hand, the pre-trip models have been estimated on two distinct samples, differing in the typical length of the trips involved. The heterogeneity of these samples has been brought to light as many socio-economic characteristics play a significant role. The model for long-distance trips is of particular interest as it has been shown that traffic information play an important role in travel decisions, as it is also the case for public transportation. This role can be correctly captured and predicted and the model can thus be used to design appropriate focused policies for long-distance trips performed in Switzerland. The work presented in this chapter constitutes a first step toward the development of Dynamic Traffic Management Systems in which it is of major importance to apprehend the impact of Intelligent Transportation Systems on travelers' behavior. From a system design point of view, the most notable conclusions of our study are linked to the willingness of the respondents to act when informed and to the impact of errors in the information.

## 6.2 Future research

In this section we outline future developments which have been inspired by the research achieved during this thesis. We make the distinction between possible algorithmic developments and future applications of the algorithms and models we propose.

### 6.2.1 Optimization viewpoint

One of the main interests for future work consists in generalizing the ideas presented in Chapters 2 and 3 to constrained nonlinear optimization.

This generalization is first motivated by applications as we are interested in solving constrained maximum likelihood problems arising when estimating advanced discrete choice models requiring non trivial constraints in order to get meaningful values of parameters as well as to get an identifiable model. In this context, it is necessary to develop specific algorithms to identify the singularity issues and to correctly perform the estimation when non trivial constraints are imposed on the parameters. Also, it remains of major importance to be able to identify a global minimum of the model maximum likelihood estimation when constraints are present.

From the theoretical point of view, singular constrained optimization is also very interesting. We have seen that a singularity in a unconstrained nonlinear optimization comes from a flat curvature in the vicinity of a local minimum, violating one of the major assumptions on the objective function in order to guarantee the fast local convergence of methods. In the constrained case, there may be another source of singularity, namely when a constraint qualification condition is not satisfied at a local minimum (for instance, the assumption of linear independence of the constraints gradients). It is interesting to develop algorithms able to efficiently solve problems for which classical assumptions for convergence of standard methods are violated. Actually, the case of possible violation of standard constraint qualifications is starting to be investigated in the literature of constrained optimization (see, for instance, Wright, 2002, Wright, 2003). Recently, Izmailov and Solodov (2004) proposed a singular-value decomposition approach in this context. The motivation for considering such irregular cases comes from various problems, where either standard constraints qualifications are inherently violated or constraints tend to be degenerate or nearly (that is numerically) degenerate. Of interest are both theoretical properties of irregular problems as well as convergence of optimization algorithms applied to such problems and, most importantly, possible modifications of the algorithms to improve robustness and efficiency.

An almost straightforward extension of the ideas presented in Chapters 2 and 3 is to combine these ideas in order to get an algorithmic method able to identify a global minimum of a singular unconstrained nonlinear problem. The way to achieve this goal will be to replace the trust-region algorithm proposed in the VNS algorithm of Chapter 3 by new trust-region based methods described in Chapter 2 which can efficiently deal with singularities in the objective function.

Some algorithmic adaptations of the VNS proposed in Chapter 3 should be worth investigating. Indeed, we believe that we could have a better estimation of convergence basins of already encountered local minima by keeping also other previous iterates and not only local minima as it is the case in the proposed version. Defining the number of generated neighbors $p$ as dynamic from iteration to iteration of the VNS might also be interesting to investigate. Another track of development would be to incorporate the VNS presented into an Adaptive Memory Method (AMM) framework (see Rochat and Taillard, 1995) in order to improve even more the diversification inside our algorithm, and the probability of finding a global minimum.

A theoretical analysis of a globally convergent version of our generalized secant method proposed in Chapter 4 must also be performed. We conjecture that the algorithm can be tailored in order to enforce a superlinear rate of local convergence. One possible way is to design the parameters such that the method becomes asymptotically equivalent to Broyden's and, consequently, would inherit its superlinear rate of convergence. This needs to be investigated further. The ideas presented in Chapter 4 could finally be tailored to unconstrained nonlinear optimization and singular unconstrained nonlinear optimization, following the tracks described in Section 4.7.

### 6.2.2 Application viewpoint

Possible applications of the algorithmic methods presented in this thesis have been briefly discussed in Chapter 1.

The algorithms proposed to deal with singular unconstrained nonlinear optimization (see Chapter 2) and with unconstrained nonlinear global optimization (see Chapter 3) have demonstrated their superiority on existing approaches according to the intensive numerical experiments performed. Our methods have been shown to significantly reduce the number of function evaluations necessary to solve their respective problems. Most importantly, it should allow to obtain a significant gain in computational time as soon as the objective function is expensive to evaluate and dominate other computational costs of numerical algebra, which is the case in most applications related to transportation involving simulation tools. Proposed methods are thus potentially very interesting in order to estimate advanced discrete choice models which require simulation to evaluate the log-likelihood function. This is why the most interesting future path to follow from an application viewpoint is to apply our methods on real applications. More precisely, the algorithmic methods described in Chapters 2 and 3 will be implemented in the software package BIOGEME in order to estimate advanced discrete choice models. Numerical tests will be conducted on discrete choice models involving singularities issues as well as discrete choice models whose likelihood function presents several local optima. We are confident to significantly reduce the estimation time for the first class of models using the two algorithms presented in Chapter 2 while the heuristic presented in Chapter 3 should be more likely to find a global optimum for the second class of models, compared to existing local algorithms available in BIOGEME.

The behavioral models presented in Chapter 5 have also been shown to have an important potential of future application. These models have been able to correctly apprehend the impact of traffic information on drivers' behavior in terme of route and mode choice decisions. They have been consequently integrated in a simulator which permits to design and evaluate hypothetical scenarii involving traffic information provided by telematics technology. It is of major importance for the future development of Intelligent Transportation Systems and Dynamic Traffic Management Systems. The models proposed in this dissertation will allow to evaluate specific policies in such contexts.

Despite the fact that the applications which have motivated the work described in this thesis are essentially related to transportation, we are persuaded that our algorithmic approaches have a high potential for applications in a wide variety of domains.

# Appendix A

# Appendix

## Contents

## A.1 List of test functions for global optimization

### A.1.1 Branin RCOS Function (RC)

- 2 variables
- $RC(x_1, x_2) = (x_2 - (5.1/4\pi^2)x_1^2 + (5/\pi)x_1 - 6)^2 + 10(1 - (1/8\pi))\cos(x_1) + 10$
- Range of initial points: $-5 < x_1 < 10$, $0 < x_2 < 15$.
- Global minima: $(x_1^*, x_2^*) = (-\pi, 12.275), (\pi, 2.275), (9.42478, 2.475)$; $RC(x_1, x_2^*) = 0.397887$

### A.1.2 Easom Function (ES)

- 2 variables
- $ES(x_1, x_2) = -\cos(x_1)\cos(x_2)e^{-(x_1-\pi)^2-(x_2-\pi)^2}$
- Range of initial points: $-10 < x_j < 10$, $j = 1, 2$.
- Several local minima
- Global minimum: $(x_1^*, x_2^*) = (\pi, \pi); ES(x_1, x_2^*) = -1$

### A.1.3 Rastrigin Function (RT)

- 2 variables
- $RT(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$
- Range of initial points: $-1 < x_j < 1$, $j = 1, 2$.
- Many local minima
- Global minimum: $(x_1^*, x_2^*) = (0, 0); RT(x_1, x_2^*) = 0$

### A.1.4 Shubert Function (SH)

- 2 variables
- $SH(x_1, x_2) = (\sum_{j=1}^{5} j\cos((j+1)x_1 + j))(\sum_{j=1}^{5} j\cos((j+1)x_2 + j))$
- Range of initial points: $-10 < x_j < 10$, $j = 1, 2$.
- 760 local minima
- 18 global minima: $SH(x_1, x_2^*) = -186.7309$

### A.1.5 De Joung Function (DJ)

- 3 variables
- $DJ(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$
- Range of initial points: $-5 < x_j < 5$, $j = 1, 2, 3$.
- Global minimum: $(x_1^*, x_2^*, x_3^*) = (0, 0, 0); DJ(x_1, x_2^*, x_3^*) = 0$

### A.1.6 Hartmann Function ($H_{3,4}$)

- 3 variables
- $H_{3,4}(x) = -\sum_{i=1}^{4} c_i e^{-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2}$
- Range of initial points: $0 < x_j < 1$, $j = 1, 2, 3$.
- 4 local minima
- Global minimum: $x* = (0.114614, 0.555649, 0.852547)$;
  $H_{3,4}(x*) = -3.86278$

| $i$ | | $a_{ij}$ | | $c_i$ | | $p_{ij}$ | |
|---|---|---|---|---|---|---|---|
| 1 | 3.0 | 10.0 | 30.0 | 1.0 | 0.689 | 0.1170 | 0.2673 |
| 2 | 0.1 | 10.0 | 35.0 | 1.2 | 0.4699 | 0.4387 | 0.7470 |
| 3 | 3.0 | 10.0 | 30.0 | 3.0 | 0.1091 | 0.8732 | 0.5547 |
| 4 | 0.1 | 10.0 | 35.0 | 3.2 | 0.0381 | 0.5743 | 0.8828 |

### A.1.7 Hartmann Function ($H_{6,4}$)

- 6 variables
- $H_{6,4}(x) = -\sum_{i=1}^{4} c_i e^{-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2}$
- Range of initial points: $0 < x_j < 1$, $j = 1, \ldots, 6$.
- 6 local minima
- Global minimum: $x* = (0.201690, 0.150011, 0.476874, 0.275332, 0.311652, 0.657300)$;
  $H_{6,4}(x*) = -3.32237$

| $i$ | $a_{ij}$ | | | | | | $c_i$ |
|---|---|---|---|---|---|---|---|
| 1 | 10.0 | 3.0 | 17.0 | 3.50 | 1.70 | 8.00 | 1.0 |
| 2 | 0.05 | 10.0 | 17.0 | 0.10 | 8.00 | 14.00 | 1.2 |
| 3 | 3.00 | 3.50 | 1.70 | 10.0 | 17.00 | 8.00 | 3.0 |
| 4 | 17.00 | 8.00 | 0.05 | 10.00 | 0.10 | 14.00 | 3.2 |

| $i$ | $p_{ij}$ | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0.1312 | 0.1696 | 0.5569 | 0.0124 | 0.8283 | 0.5886 |
| 2 | 0.2329 | 0.4135 | 0.8307 | 0.3736 | 0.1004 | 0.9991 |
| 3 | 0.2348 | 0.1451 | 0.3522 | 0.2883 | 0.3047 | 0.6650 |
| 4 | 0.4047 | 0.8828 | 0.8732 | 0.5743 | 0.1091 | 0.0381 |

### A.1.8 Shekel Functions ($S_{4,m}$)

- 4 variables
- $S_{4,m}(x) = -\sum_{i=1}^{m}(\sum_{j=1}^{4}(x_j - a_{ij})^2 + c(i))^{-1}$
- 3 functions are considered, namely: $S_{4,5}$, $S_{4,7}$ and $S_{4,10}$
- Range of initial points: $0 < x_j < 10$, $j = 1, \ldots, 4$.
- $m$ local minima
- Global minimum: $x* = (4, 4, 4, 4)$;
  $S_{4,5}(x*) = -10.1532$, $S_{4,7}(x*) = -10.4029$ and $S_{4,10}(x*) = -10.5364$

| $i$ | $a_{ij}$ | | | | $c_i$ |
|---|---|---|---|---|---|
| 1 | 4.0 | 4.0 | 4.0 | 4.0 | 0.1 |
| 2 | 1.0 | 1.0 | 1.0 | 1.0 | 0.2 |
| 3 | 8.0 | 8.0 | 8.0 | 8.0 | 0.2 |
| 4 | 6.0 | 6.0 | 6.0 | 6.0 | 0.4 |
| 5 | 3.0 | 7.0 | 3.0 | 7.0 | 0.4 |
| 6 | 2.0 | 9.0 | 2.0 | 9.0 | 0.6 |
| 7 | 5.0 | 5.0 | 3.0 | 3.0 | 0.3 |
| 8 | 8.0 | 1.0 | 8.0 | 1.0 | 0.7 |
| 9 | 6.0 | 2.0 | 6.0 | 2.0 | 0.5 |
| 10 | 7.0 | 3.6 | 7.0 | 3.6 | 0.5 |

### A.1.9 Rosenbrock Function ($R_n$)

- $n$ variables with $n = 2, 5, 10, 50, 100$
- $R_n(x) = \sum_{j=1}^{n-1} (100(x_j^2 - x_{j+1})^2 + (x_j - 1)^2)$
- Range of initial points: $-5 < x_j < 10$, $j = 1, 2, \dots, n$
- Global minimum: $x^* = (1, \dots, 1)$, $R_n(x^*) = 0$

### A.1.10 Zakharov Function ($Z_n$)

- $n$ variables with $n = 2, 5, 10, 50$
- $Z_n(x) = \sum_{j=1}^{n} x_j^2 + (\sum_{j=1}^{n} 0.5jx_j)^2 + (\sum_{j=1}^{n} 0.5jx_j)^4$
- Range of initial points: $-5 < x_j < 10$, $j = 1, 2, \dots, n$
- Global minimum: $x^* = (0, \dots, 0)$, $R_n(x^*) = 0$

### A.1.11 Hump Function (HM)

- 2 variables
- $HM(x_1, x_2) = 1.0316285 + 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
- Range of initial points: $-5 < x_j < 5$, $j = 1, 2$.
- Global minima: $(x_1^*, x_2^*) = (0.0898, -0.7126).(-0.0898, 0.7126)$; $HM(x_1, x_2^*) = 0$

### A.1.12 Griewank Function ($GR_n$)

- $n$ variables with $n = 6, 10$
- $GR_n(x) = \sum_{j=1}^{n} x_j^2/4000 - \prod_{j=1}^{n} \cos(x_j/\sqrt{j}) + 1$
- Range of initial points: $-10 < x_j < 10$, $j = 1, 2, \dots, n$
- Many local minima
- Global minimum: $x^* = (0, \dots, 0)$, $GR_n(x^*) = 0$

### A.1.13 Colville Function (CV)

- 4 variables
- $CV(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$
- Range of initial points: $-10 < x_j < 10$, $j = 1, \dots, 4$
- Global minimum: $x^* = (1, 1, 1, 1)$, $CV(x^*) = 0$

### A.1.14 Dixon Function (DX)

- 10 variables
- $DX(x) = (1 - x_1)^2 + (1 - x_{10})^2 + \sum_{j=1}^{9} (x_j^2 - x_{j+1})^2$
- Range of initial points: $-10 < x_j < 10$, $j = 1, \dots, 10$
- Global minimum: $x^* = (1, \dots, 1)$, $DX(x^*) = 0$

### A.1.15 Martin&Gaddy Function (MG)

- 2 variables
- $MG(x) = (x_1 - x_2)^2 + (\frac{x_1 + x_2 - 10}{3})^2$
- Range of initial points: $-20 < x_j < 20$, $j = 1, 2$
- Global minimum: $x^* = (5, 5)$, $MG(x^*) = 0$

## A.2 Systems of linear equations

We have tested three linear problems of the form $Ax = b$ in the numerical experiments conducted in Chapter 4. They have been designed to challenge the tested algorithms.

1. For the first, the matrix $A$ is the Hilbert matrix, and vector $b$ is composed of all ones.
2. The second problem is based on the matrix $A$ such that $a_{ij} = j$ if $i + j = n + 1$, and $a_{ij} = 0$ otherwise. All entries of the right-hand side $b$ are -10. Its structure is designed so that the identitiy matrix is a poor approximation.
3. The third problem is based on a Vandermond matrix $A(v)$ with $v = (-1, -2, \dots, -n)$. All entries of the right-hand side $b$ are -1.

The starting point for all those problems is $x = (1, \dots, 1)^T$.

# Bibliography

Androulakis, I. P., Maranas, C. D. and Floudas, C. A. (1995). $\alpha$BB: A global optimization method for general constrained nonconvex problems, *Journal of Global Optimization* **7**(4): 337–363.

Battiti, R. and Tecchiolli, G. (1996). The continuous reactive tabu search: Blending combinatorial optimization and stochastic search for global optimization, *Annals of Operations Research* **63**: 153–188.

Ben-Akiva, M. and Bierlaire, M. (2003). Discrete choice models with applications to departure time and route choice, *in* R. Hall (ed.), *Handbook of Transportation Science, Second Edition*, Kluwer Academic Publishers, Boston/Dordrecht/London, pp. 7–37.

Ben-Akiva, M., Bierlaire, M., Burton, D., Koutsopoulos, H. and Mishalani, R. (2001). Network state estimation and prediction for real-time transportation management applications, *Networks and Spatial Economics* **1**(3/4): 293–318.

Ben-Akiva, M. E. (1973). *Structure of passenger travel demand models*, PhD thesis, Department of Civil Engineering, MIT, Cambridge, Ma.

Ben-Akiva, M. E. and Lerman, S. R. (1985). *Discrete Choice Analysis: Theory and Application to Travel Demand*, MIT Press, Cambridge, Ma.

Ben-Akiva, M., Koutsopoulos, H. N., Mishalani, R. and Yang, Q. (1997). Simulation laboratory for evaluating dynamic traffic management systems, *Journal of Transportation Engineering* **123**(4): 283–289.

Benson, H. Y., Vanderbei, R. J. and Shanno, D. F. (2002). Interior-point methods for nonconvex nonlinear programming: Filter methods and merit functions, *Computational Optimization and Applications* **23**(2): 257–272.

Bertsekas, D. P. (1999). *Nonlinear Programming*, 2nd edn, Athena Scientific, Belmont.

Bhat, C. (2001). Quasi-random maximum simulated likelihood estimation of the mixed multinomial logit model, *Transportation Research Part B* **35**(7): 677–693.

Bhat, C. R. and Guo, J. Y. (2004). A mixed spatially correlated logit model: Formulation and application to residential choice modeling, *Transportation Research Part B: Methodological* **38**(2): 147–168.

Bierlaire, M. (2002). The network GEV model, *Proceedings of the 2nd Swiss Transportation Research Conference*, Ascona, Switzerland.

Bierlaire, M. (2003). BIOGEME: a free package for the estimation of discrete choice models, *Proceedings of the 3rd Swiss Transportation Research Conference*, Ascona, Switzerland. www.strc.ch.

Bierlaire, M. (2005). An introduction to BIOGEME version 1.4. biogeme.epfl.ch.

Bierlaire, M. (2006a). *Introduction à l'optimisation différentiable*, Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland.

Bierlaire, M. (2006b). A theoretical analysis of the cross-nested logit model, *Annals of operations research* **144**(1): 287–300.

Bierlaire, M. and Crittin, F. (2003). Generalization of secant methods for solving systems of nonlinear equations, *Proceedings of the 3rd Swiss Transportation Research Conference*, Ascona, Switzerland.

Bierlaire, M. and Crittin, F. (2004). An efficient algorithm for real-time estimation and prediction of dynamic OD tables, *Operations Research* **52**(1): 116–127.

Bierlaire, M. and Crittin, F. (2006). Solving noisy large scale fixed point problems and systems of nonlinear equations, *Transportation Science* **40**(1): 44–63.

Bierlaire, M., Crittin, F. and Thémans, M. (to appear). A multi-iterate method to solve systems of nonlinear equations, *European Journal of Operational Research* **183**(1): 20–41.

Bierlaire, M. and Thémans, M. (2006a). Dealing with singularities in nonlinear unconstrained optimization, *Technical report TRANSP-OR-061207*, Transport and Mobility Laboratory - Ecole Polytechnique Fédérale de Lausanne Technical report, Lausanne, Switzerland.

Bierlaire, M. and Thémans, M. (2006b). Développement de modèles suisses pour la prédiction de la demande en transport pour des applications en temps réel, *Technical Report OFROU VSS 1135 novembre 2004*, Operations Research Group ROSO, Ecole Polytechnique Fédérale de Lausanne.

Bierlaire, M., Thémans, M. and Axhausen, K. (2006). Analysis of driver's response to real-time information in Switzerland, *European Transport* **34**: 21–41.

Bierlaire, M., Thémans, M. and Zufferey, N. (2007). A new heuristic for nonlinear global optimization, *Technical report TRANSP-OR-070430*, Transport and Mobility Laboratory - Ecole Polytechnique Fédérale de Lausanne Technical report, Lausanne, Switzerland.

Bottom, J. (2000). *Consistent Anticipatory Route Guidance*, Cambridge, ma., Massachusetts Institute of Technology.

Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations, *Mathematics of Computation* **19**: 577–593.

Bundesamt für Raumentwicklung and Bundesamt für Statistik (2001). Mobilität in der schweiz, ergebnisse des mikrozensus 2000 zum verkehrsverhalten, Bern and Neuenburg.

Byrd, R., Nocedal, J. and Schnabel, R. B. (1994). Representation of quasi-Newton matrices and their use in limited memory methods, *Mathematical Programming* **63**: 129–136.

Chatterjee, K., Hounsell, N. B., Firmin, P. E. and Bonsall, P. W. (2002). Driver response to variable message sign information in London, *Transportation Research Part C* **10**(2): 149–169.

Chelouah, R. and Siarry, P. (1997). Enhanced simulated annealing for globally minimizing functions of many continuous variables, *ACM Transactions on Mathematical Software* **23**(2): 209–228.

Chelouah, R. and Siarry, P. (2000a). A continuous genetic algorithm designed for the global optimization of multimodal functions, *Journal of Heuristics* **6**: 191–213.

Chelouah, R. and Siarry, P. (2000b). Tabu search applied to global optimization, *European Journal of Operations Research* **123**: 256–270.

Chelouah, R. and Siarry, P. (2003). Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions, *European Journal of Operational Research* **148**: 335–348.

Conn, A. R., Gould, N. I. M. and Toint, P. (2000). *Trust region methods*, MPS–SIAM Series on Optimization, SIAM.

Conn, A. R., Gould, N. I. M. and Toint, P. L. (1992). LANCELOT*: a Fortran package for large-scale nonlinear optimization (Release A)*, number 17 in *Springer Series in Computational Mathematics*, Springer Verlag, Heidelberg, Berlin, New York.

Conquest, L., Spyridakis, J., Haselkorn, M. and Barfield, W. (1993). The effect of motorist information on commuter behavior: Classification of drivers into commuter groups, *Transportation Research Part C* .

Crittin, F. (2004). *New algorithmic methods for real-time transportation problems*, PhD thesis, Ecole Polytechnique Fdrale de Lausanne, Lausanne, Switzerland.

Daly, A. and Bierlaire, M. (2006). A general and operational representation of generalised extreme value models, *Transportation Research Part B* **40**(4): 285–305.

Davidon, W. C. (1991). Variable metric method for minimization, *SIAM Journal on Optimization* **1**: 1–17.

Decker, D. W. and Kelley, C. T. (1980). Newton's method at singular points ii., *SIAM Journal on Numerical Analysis* **17**(3): 465–471.

Decker, D. W., Kelley, H. B. and Kelley, C. T. (1983). Convergence rates for Newton's method at singular points, *SIAM Journal on Numerical Analysis* **20**(2): 296–314.

Dekkers, A. and Aarts, E. (1991). Global optimization and simulated annealing, *Mathematical Programming* **50**: 367–393.

Dennis, J. E. and Schnabel, R. B. (1996). *Numerical methods for unconstrained optimization and nonlinear equations*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa.

Dia, H. (2002). An agent-based approach to modelling driver route choice behaviour under the influence of real-time information, *Transportation Research Part C* **10**(5–6): 331–349.

Diniz-Ehrhardt, M. A., Gomes-Ruggiero, M. A., Lopes, V. L. R. and Martínez, J. M. (2003). Discrete Newton's method with local variations for solving large-scale nonlinear systems, *Optimization* **52**: 417–440.

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

Dorigo, M. and Blum, C. (2005). Ant colony optimization theory: A survey, *Theoretical Computer Science* **344(2-3)**: 243–278.

Dorsey, R. E. and Mayer, W. J. (1995). Genetic algorithms for estimation problems with multiple optima, nondifferentiability, and other irregular features, *Journal of Business & Economic Statistics* **13**(1): 53–66.

Eaton, J. W. (1997a). GNU Octave: a high level interactive language for numerical computations, `www.octave.org`.

Eaton, J. W. (1997b). GNU *Octave Manual*, Series on Optimization, Network Theory Limited, Bristol, United Kingdom.

Eyert, V. (1996). A comparative study on methods for convergence acceleration of iterative vector sequences, *Journal of computational physics* **124**: 271–285.

Fletcher, R. (1970). A new approach to variable metric algorithms, *Computer Journal* **13**: 317–322.

Fletcher, R. and Leyffer, S. (1997). Nonlinear programming without a penalty function, *Technical Report NA/171*, Department of Mathematics, University of Dundee, Scotland.

Fletcher, R. and Leyffer, S. (2002). Nonlinear programming without a penalty function, *Mathematical Programming* **91**.

Fletcher, R. and Leyffer, S. (2003). Filter-type algorithms for solving systems of algebraic equations and inequalities, *in* G. D. Pillo and A. Murli (eds), *High Performance Algorithms and Software in Nonlinear Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Fletcher, R., Leyffer, S. and Toint, P. L. (2002). On the global convergence of a filter-SQP algorithm, *SIAM Journal on Optimization* **13**(1): 44–59.

Floudas, C., Pardalos, P., Adjiman, C., Esposito, W., Gumus, Z., Harding, S., Klepeis, J., Meyer, C. and Schweiger, C. (1999). *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic Publishers.

Ford, J. A. (1999). A survey of multi-step quasi-Newton methods, *Proceedings of the International Conference on Scientific Computations*, Beirut, Lebanon.

Ford, J. A. and Moghrabi, I. A. (1997). Alternating multi-step quasi-Newton methods for unconstrained optimization, *Journal of Computational and Applied Mathematics* **82**: 105–116.

Franzè, F. and Speciale, N. (2001). Tabu search applied to global optimization, *International Journal for Numerical Methods in Engineering* **50**: 665–680.

Glover, F. (1986). Future paths for integer programming and linkage to artificial intelligence, *Computers & Operations Research* **13**: 533–549.

Glover, F. (1994). Tabu search for nonlinear and parametric optimization (with links to genetic algorithms), *Discrete Applied Mathematics* **49**: 231–255.

Glover, F. (1998). A template for scatter search and path relinking, *Lecture Notes on Computer Science* **1363**: 3 – 51.

Goffe, W. L., Ferrier, G. D. and Rogers, J. (1992). Simulated annealing: An initial application in econometrics, *Computer Science in Economics and Management* **5**: 133–146.

Goffe, W. L., Ferrier, G. D. and Rogers, J. (1994). Global optimization of statistical functions with simulated annealing, *Journal of Econometrics* **60**: 65–99.

Goldfarb, D. (1970). A family of variable metric methods derived by variational means, *Mathematics of Computation* **24**: 23–26.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*, third edn, Johns Hopkins University Press, Baltimore and London.

Gomes-Ruggiero, M. A., Martínez, J. M. and Moretti, A. C. (1992). Comparing algorithms for solving sparse nonlinear systems of equations, *SIAM J. Sci. Stat Comp.* **13**(2): 459–483.

Gould, N. I. M., Leyffer, S. and Toint, P. L. (2005). A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares, *SIAM Journal on Optimization* **15**(1): 17–38.

Gould, N. I. M., Orban, D. and Toint., P. L. (2002). General CUTEr documentation, *Technical Report TR/PA/02/13*, CERFACS.

Gould, N. I. M., Sainvitu, C. and Toint, P. L. (2006). A filter-trust-region method for unconstrained optimization, *SIAM Journal on Optimization* **16**(2): 341–357.

Gragg, W. and Stewart, G. (1976). A stable variant of the secant method for solving nonlinear equations, *SIAM Journal on Numerical Analysis* **13**: 889–903.

Griewank, A. O. (1985). On solving nonlinear equations with simple singularities or nearly singular solutions, *SIAM Review* **27**(4): 537–563.

Griewank, A. O. and Osborne, M. R. (1983). Analysis of Newton's method at irregular singularities, *SIAM Journal on Numerical Analysis* **20**(4): 747–773.

Hansen, E. R. and Walster, G. W. (2003). *Global optimization using interval analysis*, CRC Press.

Hansen, P. (1986). The steepest ascent mildest descent heuristic for combinatorial programming, *Congress on Numerical Methods in Combinatorial Optimization*, Capri, Italy.

Hedar, A. and Fukushima, M. (2002). Hybrid simulated annealing and direct search methods for nonlinear unconstrained global optimization, *Optimization Methods and Software* **17**: 891–912.

Hedar, A. and Fukushima, M. (2004). Heuristic pattern search and its hybridization with simulated annealing for nonlinear global optimization, *Optimization Methods and Software* **19**: 291–308.

Hedar, A. and Fukushima, M. (2006). Tabu search directed by direct search methods for nonlinear global optimization, *European Journal of Operations Research* **170**: 329–349.

Henrion, D. and Lasserre, J.-B. (2003). Gloptipoly: Global optimization over polynomials with Matlab and SeDuMi, *ACM Transactions on Mathematical Software* **29**(2): 165–194.

Hertz, A., Plumettaz, M. and Zufferey, N. (2006). Variable space search for graph coloring, *submitted for publication* .

Hess, S., Bierlaire, M. and Polak, J. (2004). Development and application of a mixed cross-nested logit model, *Proceedings of the XXIth European Transport Conference*, Strasbourg.

Hess, S., Bierlaire, M. and Polak, J. (2005a). Capturing correlation and taste heterogeneity with mixed GEV models, *in* R. Scarpa and A. Alberini (eds), *Application of simulation methods in environmental and resource economics*, Vol. 6 of *The Economics of Non-Market Goods and Resources*, Springer.

Hess, S., Bierlaire, M. and Polak, J. (2005b). Estimation of value of travel-time savings using mixed logit models, *Transportation Research Part A* **39**(3): 221–236.

Hess, S., Bierlaire, M. and Polak, J. (2005c). Mixed GEV modelling of mode choice, *Proceedings of the 5th Swiss Transportation Research Conference*. www.strc.ch.

Hess, S., Bierlaire, M. and Polak, J. (to appear). A systematic comparison of continuous and discrete mixture models, *European Transport* . Accepted for publication.

Horst, R. and Thoai, N. V. (1999). DC programming: Overview, *Journal of Optimization Theory and Applications* **103**(1): 1–43.

Izmailov, A. F. and Solodov, M. V. (2004). Newton-type methods for optimization problems without constraint qualifications, *SIAM Journal on Optimization* **15**(1): 210–228.

Izmailov, A. and Solodov, M. (2002). Superlinearly convergent algorithms for solving singular equations and smooth reformulations of complementarity problems, *SIAM Journal on Optimization* **13**: 386–405.

Johnson, D. D. (1988). Modified Broyden's method for accelerating convergence in self-consistent calculations, *Physical Review B* **38**(18): 12807–12813.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems, *J. of Basic Eng., Trans. ASME, Series D* **82**(1): 33–45.

Kelley, C. T. (2003). *Solving nonlinear equations with Newton's method*, SIAM, Philadelphia, Pa.

Khattak, A. J., Yim, Y. and Prokopy, L. S. (2003). Willingness to pay for travel information, *Transportation Research Part C* **11**(2): 137–159.

Khattak, A., Polydoropoulou, A. and Ben-Akiva, M. (1996). Modeling revealed and stated pretrip travel response to advanced traveler information systems, *Transportation Research Record* **1537**: 46–54.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983). Optimization by simulated annealing, *Science* **220**: 671–680.

Klepeis, J. L., Pieja, M. J. and Floudas, C. A. (2003). Hybrid global optimization algorithms for protein structure prediction: Alternating hybrids, *Biophysical Journal* **84**: 869–882.

Koenig, A., Abay, G. and Axhausen, K. (2004). Zeitkostenansaetze im personenverkehr, *Schriftenreihe 1065*, Bundesamt für Strassen, UVEK, Bern. final report for SVI 2001/534.

Lasserre, J. B. (2001). Global optimization with polynomials and the problem of moments, *SIAM Journal on Optimization* **11**(3): 796–817.

Lasserre, J. B. (2004). Solving nonconvex optimization problems, *IEEE Control Systems Magazine* **24**: 72–83.

Le, T. H. A. and Pham, D. T. (2005). The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems, *Annals of Operations Research* **133**: 23–46.

Lin, Y. and Stadtherr, M. A. (2004). Advances in interval methods for deterministic global optimization in chemical engineering, *Journal of Global Optimization* **29**(3): 281–296.

Locatelli, M. (2000). Simulated annealing algorithms for continuous global optimization: Convergence conditions, *Journal of Optimization Theory and Applications* **104**(1): 121–133.

Mahmassani, H. S. and Liu, Y.-H. (1999). Dynamics of commuting decision behaviour under advanced traveller information systems, *Transportation Research Part C* **7**(2–3): 91–107.

Manski, C. and Lerman, S. (1977). The estimation of choice probabilities from choice-based samples, *Econometrica* **45**: 1977–1988.

Manski, C. and McFadden, D. (1981). Alternative estimators and sample designs for discrete choice analysis, *in* C. Manski and D. McFadden (eds), *Structural analysis of discrete data with econometric application*, MIT Press, Cambridge, Mass.

Martínez, J. M. (1979). Three new algorithms based on the sequantial secant method, *BIT* **19**: 236–243.

Martínez, J. M. (2000). Practical quasi-Newton methods for solving nonlinear systems, *Journal of Computational and Applied Mathematics* **124**: 97–122.

Martínez, J. M. and Ochi, L. S. (1982). Sobre dois métodos de Broyden, *Matemática Aplicada e Computacional* **1**: 135–143.

McFadden, D. (1978). Modelling the choice of residential location, *in* A. Karlquist *et al.* (ed.), *Spatial interaction theory and residential location*, North-Holland, Amsterdam, pp. 75–96.

McFadden, D. and Train, K. (2000). Mixed MNL models for discrete response, *Journal of Applied Econometrics* **15**(5): 447–470.

Mladenovic, N., Drazic, M., Kovacevic-Vujcic, V. and Cangalovic, M. (2006). General variable neighborhood search for the continuous optimization, *Les Cahiers du GERAD* **48**.

Mladenovic, N. and Hansen, P. (1997). Variable neighborhood search, *Computers & Operations Research* **24**(11): 1097–1100.

Moghrabi, I. (1993). *Multi-step quasi-Newton methods for optimization*, PhD thesis, University of Essex, United Kingdom.

Moré, J. J., Garbow, B. S. and Hillstrom, K. E. (1981). Testing unconstrained optimization software, *ACM Transactions on Mathematical Software* **7**(1): 17–41.

Neveu, B., Trombettoni, G. and Glover, F. (2004). ID Walk: a candidate list strategy with a simple diversification device, *in* M. Wallace (ed.), *Principles and Practice of Constraint Programming*, Springer, pp. 423–437.

Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*, Operations Research, Springer Verlag, New-York.

Ortega, J. M. and Rheinboldt, W. C. (1970). *Iterative solution of nonlinear equations in several variables*, Academic Press, New York.

Paige, C. C. and Saunders, M. A. (1982). LSQR: an algorithm for sparse linear equations and sparse least squares, *ACM Transactions on Mathematical Software* **8**: 43–71.

Papola, A. (2004). Some developments on the cross-nested logit model, *Transportation Research B* **38**(9): 833–851.

Renders, J.-M. and Flasse, S. P. (1996). Hybrid methods using genetic algorithms for global optimization, *IEEE Transactions on Systems, Man and Cybernetics - Part B:Cybernetics* **26**(2): 243–258.

Rochat, Y. and Taillard, E. (1995). Probabilistic diversification and intensification in local search for vehicle routing, *Journal of Heuristics* **1**: 147–167.

Saad, Y. and Schultz, M. (1986). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on Numerical Analysis* **7**: 856–869.

Sandgren, E. and Cameron, T. M. (2002). Robust design optimization of structures through consideration of variation, *Computers & Structures* **80**(20-21): 1605–1613.

Sartenaer, A. (1997). Automatic determination of an initial trust region in nonlinear programming, *SIAM Journal on Scientific Computing* **18**(6): 1788–1803.

Schnabel, R. B. and Eskow, E. (1991). A new modified Cholesky factorization, *SIAM Journal on Scientific and Statistical Computing* **11**: 1136–1158.

Schnabel, R. B. and Eskow, E. (1999). A revised modified Cholesky factorization, *SIAM Journal on Optimization* **9**: 1135–1148.

Schnabel, R. B. and Frank, P. D. (1984). Tensor methods for nonlinear equations, *SIAM Journal on Numerical Analysis* **21**(5): 815–843.

Schnabel, R. and Chow, T.-T. (1991). Tensor methods for unconstrained optimization using second derivatives, *SIAM Journal on Optimization* **1**(3): 293–315.

Spedicato, E. and Huang, Z. (1997). Numerical experience with Newton-like methods for nonlinear algebraic systems, *Computing* **58**: 69–99.

Srinivasan, K. K. and Mahmassani, H. S. (2003). Analyzing heterogeneity and unobserved structural effects in route-switching behavior under ATIS: a dynamic kernel logit formulation, *Transportation Research Part B* **37**(9): 793–814.

Steihaug, T. (1983). The conjugate gradient method and trust regions in large scale optimization, *SIAM Journal on Numerical Analysis* **20**(3): 626–637.

Teh, Y. S. and Rangaiah, G. P. (2003). Tabu search for global optimization of continuous functions with application to phase equilibrium calculations, *Computers and Chemical Engineering* **27**(11): 1665–1679.

Thémans, M. and Bierlaire, M. (2006). Solving singularity issues in the estimation of econometric models, *6th STRC Swiss Transport Research Conference*, Monte Verita, Ascona, Switzerland. www.strc.ch.

Toint, P. L. (1981). Towards an efficient sparsity exploiting Newton method for minimization, *in* I. S. Duff (ed.), *Sparse Matrices and Their Uses*, Academic Press, London, pp. 57–88.

Train, K. (2003). *Discrete Choice Methods with Simulation*, Cambridge University Press. http://emlab.berkeley.edu/books/choice.html.

Vanderbilt, D. and Louie, S. G. (1984). Total energies of diamond (111) surface reconstructions by a linear combination of atomic orbitals method, *Physical Review B* **30**(10): 6118–6130.

Vaz, A. I. F. and Vicente, L. N. (to appear). A particle swarm pattern search method for bound constrained global optimization, *Journal of Global Optimization* . Accepted for publication.

Vovsha, P. (1997). Cross-nested logit model: an application to mode choice in the Tel-Aviv metropolitan area, *Transportation Research Record* **1607**: 6–15.

Wächter, A. and Biegler, L. T. (2001). Global and local convergence of line search filter methods for nonlinear programming, *Technical report*, CAPD Technical Report B-01-09, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, Pa, USA.

Wächter, A. and Biegler, L. T. (2005a). Line search filter methods for nonlinear programming: Local convergence, *SIAM Journal on Optimization* **16**(1): 32–48.

Wächter, A. and Biegler, L. T. (2005b). Line search filter methods for nonlinear programming: Motivation and global convergence, *SIAM Journal on Optimization* **16**(1): 1–31.

Walker, J. L. (2001). *Extended discrete choice models: integrated framework, flexible error structures, and latent variables*, PhD thesis, Massachusetts Institute of Technology.

Wardman, M., Bonsall, P. W. and Shires, J. D. (1997). Driver response to variable message signs: a stated preference investigation, *Transportation Research Part C* 5(6): 389–405.

Wen, C.-H. and Koppelman, F. S. (2001). The generalized nested logit model, *Transportation Research Part B* **35**(7): 627–641.

Williams, H. (1977). On the formation of travel demand models and economic measures of user benefit, *Environment and Planning* **9A**: 285–344.

Wolfe, P. (1959). The secant method for solving nonlinear equations, *Communications ACM* **12**: 12–13.

Wright, S. J. (2002). Modifying SQP for degenerate problems, *SIAM Journal on Optimization* **13**(2): 470–497.

Wright, S. J. (2003). Constraint identification and algorithm stabilization for degenerate nonlinear programs, *Mathematical Programming* **95**: 137–160.

Zhao, S. (1996). *A study on route choice behavior in response to traffic information using drivers' perception*, PhD thesis, University of Tokyo, Japan.

# Curriculum Vitae

# Michaël Thémans

## Address

Ch. du Suchet 7A  
1350 ORBE (VD)  
Switzerland  
Phone: +41 21 693 25 32  
E-mail: michael.themans@epfl.ch  
Homepage: `http://roso.epfl.ch/mt`

## Identity

Gender: male  
Date of birth: 17th August 1980  
Place of birth: Namur (Belgium)  
Nationality: Belgian

## Education

| | |
|---|---|
| 2002-2007 | *PhD Thesis in Mathematics (numerical optimization)*<br>**École Polytechnique Fédérale de Lausanne (EPFL)** |
| 2000-2002 | *Licence en Sciences Mathématiques*<br>*Thesis: Modélisation et simulation du flux d'une ligne d'embouteillage*<br>**University of Namur (FUNDP), Belgium** |
| 1998-2000 | *Candidature en Sciences Mathématiques*<br>**University of Namur (FUNDP), Belgium** |

## Professional experience

| | |
|---|---|
| 2002-2007 | **EPFL**<br>*Research assistant of Prof. Th. M. Liebling (chair of Operations Research) and Prof. M. Bierlaire (Transport and Mobility Laboratory)*<br>*Supervision of semester and diploma projects* |
| 2004-2007 | **EPFL - Advanced Continuing Education Course**<br>*Assistant for the course*<br>*"Discrete Choice Analysis: Predicting Demand and Market Shares"* |