UNIVERSITÀ
DEGLI STUDI
FIRENZE

Università degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione (DINFO)
Corso di Dottorato in Ingegneria dell'Informazione
Curriculum: Automatica, Ottimizzazione e Sistemi Complessi

Nonmonotone Techniques for Smooth Optimization

Candidate
Leonardo Galli

Supervisors
Prof. Marco Sciandrone

Prof. Fabio Schoen

PhD Coordinator
Prof. Fabio Schoen

ciclo XXXII, 2016-2020

Università degli Studi di Firenze, Dipartimento di Ingegneria dell'Informazione (DINFO).

Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Information Engineering.

A tutte le mie famiglie

Acknowledgments

I would like to acknowledge the efforts and input of my supervisors, Prof. Marco Sciandrone and Prof. Fabio Schoen, and all my colleagues of the Global Optimization Laboratory (LABGOL) who were of great help during my research. In particular my thanks go to Dr. Alessandro Galligari who collaborated on the main parts of my research work. I would like to thank also Prof. Christian Kanzow and Prof. Chih-Jen Lin for their kind support and hospitality during my stays at Würzburg, Los Angeles and Taipei.

The focus of this thesis is the study and the application of nonmonotone strategies. These techniques are basically introduced to improve numerical results of existing optimization algorithms. Their first aim is that of relaxing the monotone requirement imposed by the globalization techniques. In fact, these monotone conditions might slow down the convergence rate of inherently nonmonotone optimization methods. This relaxation must not harm global convergence results.

In this thesis we apply nonmonotone strategies to both line search and trust-region globalization techniques. We first considered Generalized Nash Equilibrium Problems (GNEPs) and their KKT reformulation into a highly nonlinear constrained smooth system of equations. In order to obtain global and fast local convergence, we take into account an existing trust-region method that is locally superlinear under an error bound condition only. A nonmonotone strategy has been applied, showing that the resulting algorithm performs significantly better than the original one. Global convergence properties have been proved for the new algorithm, while superlinear convergence is directly inherited from the existing method. The resulting algorithm is competitive with a standard software for nonlinear equations, not only on GNEPs, but also on quasi-variational inequalities.

The second contribution of this thesis is the development of a framework for nonmonotone line search based decomposition methods. This is the first time in which nonmonotonicity is combined with decomposition methods for general constrained problems. Note that the choice of the direction and the line search are not fixed in advance, in fact the framework proves convergence for all those combinations of directions and line searches that are able to satisfy some mild assumptions. A specific realization of this abstract algorithm has been implemented in two versions, monotone and nonmonotone. The two algorithms have been compared on a set of network equilibrium problems. Also on this application, the nonmonotone version outperformed its monotone counterpart both on the total number of iterations and the function evaluations.

In the end, a new family of nonmonotone techniques is proposed to build algorithms that are able to control the amount of nonmonotonicity introduced in each of the phases of the optimization procedure. This tool might be very helpful to understand in which combination of methods, problems and phases is more important to apply a nonmonotone strategy.

Contents

Abbreviations

AD Alternating Direction

BB Barzilai-Borwein

CG Conjugate Gradient

DF Derivative Free

GN Gauss Newton

GNEP Generalized Nash Equilibrium Problem

KKT Karush-Kuhn-Tucker

L-BFGS Limited-Broyden Fletcher Goldfarb Shanno

LCP Linear Complementarity Problem

LM Levenberg-Marquardt

NCP Nonlinear Complementarity Problem

NE Network Equilibrium

NLE NonLinear Equation

NS Non-Smooth

PRP Polak-Ribiére-Polyak

QN Quasi Newton

QP Quadratic Programming

QVI Quasi-Variational Inequalities

SANE Spectral Algorithm for Nonlinear Equations

SPG Spectracl Projected Gradient

SQP Sequential Quadratic Programming

SS Semi-Smooth

SVM Support Vector Machine

TN Truncated Newton

TR Trust-Region

Chapter 1

Introduction

Nonmonotone techniques are valuable tools for improving numerical results of existing optimization methods. In particular, they are very well suited for all those cases in which the trial step is showed to perform well when applied directly, but the same good performances cannot be maintained when a monotone globalization techniques is applied. On the other hand, without a globalization strategy it would not be possible to prove global convergence theorems for those methods. A nonmonotone technique is (generally) a small modification on the globalization procedure whose aim is that of accepting also trial steps that would result in an increase of the function value. Because of that, the sequence of function values $\{f(x^k)\}$ usually would no more be monotonically decreasing. There are in fact some specific problems, methods or optimization phases on which nonmonotonicity can be very helpful to improve the performances over their monotone counter-part.

1.1 The objective

The aim of this thesis is that of applying nonmonotone techniques to fields in which this idea was still unexplored. The majority of the optimization researchers are aware of the existence of such an extension, but there are still some fields in which this powerful tool has never been tried or still without enough effort/expertise. In general, this has happened and still happens when researchers from other areas apply optimization methods to solve problems that originate in their field. As it will be showed in Chapter 4, one example is that of network equilibrium problems. Another case is that of machine learning, in which nonmonotone techniques have been tried in the past, but most of the state-of-the-art software has never taken into account the possibility of extending their implementations to the use of nonmonotone techniques. On the other side, thanks to some useful discussion with some foreign colleagues we came to the knowledge that also between researchers that are aware of the existence of this tool, there is some skepticism on its actual value. One possible reason is the fact that nonmonotone techniques has often been applied to improve numerical results of specific methods, but not much has been done for understanding properly where they are actually more useful. For instance, there are still two important elements missing for obtaining a better comprehension of this field: a systematic literature review and a common set of problems on which validate performances of different nonmonotone techniques. Nonetheless, nonmonotone techniques can be easily implemented in all the cases in which a globalization technique is required, and this simple modification can greatly improve performances in lot of circumstances.

1.2 Contributions

In this thesis we first apply nonmonotonicity to two fields in which this idea was never exploited before: Generalized Nash Equilibrium Problems (GNEPs) and Network Equilibrium (NE) problems. In both cases the nonmonotone version of the algorithm is able to outperform its monotone counter-part. In the first case, nonmonotonicity was applied to a trust-region globalization technique, while in the second case to a line search. We thus studied and extended existing monotone proofs and methods using both the globalization techniques. The experience obtained on modeling a nonmonotone algorithm, especially from the numerical point of view, was exploited in the last of the three works. Differently from the previous projects, it mainly focus on nonmonotonicity in general. The aim is that of providing a new family of nonmonotone techniques which not only can improve existing nonmonotone performances, but might be also exploited to obtain useful insights on the optimization procedure.

In Chapter 3 one of the first trust-region nonmonotone techniques [91] is used as a tool to improve numerical results obtained by its monotone counter-part [92]. Global convergence results have been extended to the nonmonotone case. To the best of our knowledge, algorithm described in

Chapter 3, based on [46], is the first that

- exploits nonmonotonicity to solve GNEPs,
- obtains superlinear convergence for GNEPs.

In fact, GNEPs might be reformulated to obtain a constrained smooth system of equations which is often singular at the solutions. Thanks to the recent paper [62], it was anyway possible to obtain an error bound condition for GNEPs. For this reason, the method from [92] was chosen: it superlinearly solves constrained system of smooth equations only under an error bound condition. The direction employed in [92] is the convex combination of a (projected) trust-region and (projected) anti-gradient that minimizes the quadratic approximation of the merit function. For this reason it is very important to accept the new step as often as possible, especially when the trust-region direction is prevailing on the anti-gradient one, as in the late phase of the optimization procedure.

In Chapter 4 the first nonmonotone line search [55] is applied in combination with decomposition methods. This is the first time in which nonmonotonicity is theoretically studied together with a decomposition scheme for a very general class of constrained optimization problems. Based on the paper [45], this chapter extends the monotone decomposition framework [16] to the nonmonotone case. Moreover, in the case of feasible sets defined by a single equality and box constraints, global convergence is for the first time proved for a direction with only two nonzero components selected with the Gauss-Southwell rule. From the numerical point of view this is the first time in which a nonmonotone technique is applied to solve NE problems. Nonmonotonicity is again exploited to accept as often as possible the new initial step size, since in this application the Dafermos step [18] was proved to work very well in practice when applied directly. In the end, a new nonmonotone implementation is designed to obtain a consistent amount of nonmonotonicity even on large-scale problems and in presence of numerical cancellation errors.

In Chapter 5, nonmonotonicity is studied in a more general perspective. As it will be clarified below, in the field of nonmonotone techniques a practical rule that helps researchers understand where nonmonotone techniques should or should not be applied is still missing. In order to make a small step in that direction, in Chapter 5, a new family of nonmonotone techniques is proposed to better control the amount of nonmonotonicity introduced. In particular, thanks to this new idea it is possible to extract useful insights on the amount of nonmonotone effect to be introduced in the various phases of the optimization procedure. This process let us obtain benefits that go beyond those already obtained by switching from a monotone condition to a nonmonotone one. The numerical analysis have been conducted on the same set of NE problems explored in Chapter 5. The origin of this idea comes from the identification of a connection between two widely used nonmonotone techniques: [55] and [70]. This same idea was actually also exploited to define the new nonmonotone implementation in Chapter 5.

Chapter 2

State of the Art

In this chapter we will revise literature on nonmonotone techniques. These techniques have always been applied to improve numerical performances of very specific methods. In fact they have been the perfect globalization technique for lot of different fast local procedures. On the other side, this wide range of applications did not help achieving a systematic approach in the study of these techniques. For this reason, it is hard to say that a proper field of research on nonmonotone techniques is now existing. In addition, after more than thirty years from the first work on this field, lot of researchers ignore or are skeptical about the importance of nonmonotone techniques. A possible cause is the fact that nonmonotonicity is still not completely understood. In particular, it is still not clear on which problems/methods/optimization phases nonmonotonicity is more/less useful. In this literature review we focus on those papers that might be considered a proper contribution to such an ethereal field. Papers are cited in a chronological order.

- Grippo et al. [55]: The first nonmonotone technique is introduced to accept unitary step on the Newton direction in the unconstrained setting. They take inspiration from the nonmonotone watchdog technique [17], but in (2.2) they simply extend Armijo monotone technique (2.1) to accept also nonmonotone steps. In [55] they find some evidences regarding the fact that nonmonotone techniques are particularly valuable for highly nonlinear functions with narrow curved valleys and in the intermediate and in the final stages of the minimization process.

$$f(x^k + \alpha^k d^k) \leq f(x^k) + \gamma \alpha^k \nabla f(x^k)^T d^k, \tag{2.1}$$

$$f(x^k + \alpha^k d^k) \leq f^k_{max} + \gamma \alpha^k \nabla f(x^k)^T d^k, \tag{2.2}$$

where x^k, d^k and α^k respectively denotes iterate, direction and step length at iteration k, $\nabla f(x^k)$ is the gradient of f and $\gamma \in (0,1)$.

$$f^k_{max} := \max_{0 \leq j \leq \min(k,W)} f(x^{k-j}). \tag{2.3}$$

- Grippo et al. [50]: The nonmonotone technique (2.2) is for the first time applied to truncated Newton methods. They find some evidences regarding the fact that nonmonotone strategies are helpful for solving ill-conditioned problems.

- Grippo et al. [51]: They define a stabilization strategy flexible enough to take into account various different causes of inefficiency. In particular it combines the nonmonotone line search technique (2.2) with the watchdog technique of [17] and makes use of a step length ($\|x^{k+1} - x^k\|$) acceptability criterion rather than controlling the function value. A new nonmonotone strategy is introduced and f^k_{max} is replaced by something different. Condition (2.4) often introduce less nonmonotonicity than the original (2.2). On the other hand, they exploit a further relaxation of the monotonicity property: some steps can be automatically accepted, provided that they are sufficiently short. The function value is checked only every L iterations and $x^{l(j)}$ are reference iterations.

$$\begin{aligned} & f(x^k + \alpha^k d^k) \leq M^k + \gamma \alpha^k \nabla f(x^k)^T d^k \quad \text{with} \\ & M^k = \max_{0 \leq i \leq W} f(x^{l(j+1-i)}) \text{ or} \\ & M^k = \min \left\{ f^k_{max}; \frac{1}{2}(M^{j-1} + f(x^{l(j+1)})) \right\} \quad \text{or} \\ & M^k = \max \left\{ f(x^{l(j+1)}); \frac{1}{W+1} \sum_{i=0}^{W} f(x^{l(j+1-i)}) \right\} \end{aligned} \tag{2.4}$$

- Deng et al. [28]: a nonmonotone technique is for the first time developed for a trust-region method. They use (2.5) and they point out that W and Δ_{max} must be accurately chosen. They show that with the new nonmonotone method it's possible to save a considerable amount of computation for problem with narrow valley. They also give a new algorithm considering the connection with the nonmonotone strategy.

We recall here the definition of actual ($Ared^k$) and predicted ($Pred^k$) reduction,

$$Ared^k = f(x^k) - f(x^{k+1}) \qquad Pred^k = \Phi^k(0) - \Phi^k(d^k),$$

where Φ^k is a quadratic approximation of f.

$$\frac{Ared^k}{Pred^k} \geq \min\left\{\gamma, \frac{f^k_{max} - f(x^k) - \rho\Delta^k \|\nabla f(x^k)\|}{-Pred^k}\right\}, \tag{2.5}$$

where $\rho > 0$ and $\Delta^k > 0$ is the trust-region radius.

- Zhou and Tits [105]: a new nonmonotone line search mapping (2.6) is applied on sequential quadratic programming for minimax problems. In particular the decrease of the objective function is checked with respect to a quadratic term. If the condition is not satisfied using the initial step size, an arc line search is instead used to compute the step length. In this field, the event in which the line search prevent the superlinear convergence to take place is called Maratos effect. Convergence is proved and the Maratos effect avoided. In this application they use $W = 2$.

$$\begin{cases} f(x^k + \alpha^k d^k) \leq f^k_{max} + \alpha^k {d^k}^T H^k d^k & \text{if } \alpha^k_0 \text{ is accepted,} \\ f(x^k + \alpha^k d^k + {\alpha^k}^2 \tilde{d}^k) \leq f^k_{max} + \gamma\alpha^k {d^k}^T H^k d^k & \text{otherwise,} \end{cases} \tag{2.6}$$

where $\tilde{d^k}$ is a corrective direction and H is a matrix that approximates the Hessian of f.

- Ferris and Lucidi [40]: a new stabilization strategy similar to that of [51] is applied together with a nonmonotone technique to solve systems of nonlinear nonsmooth equations. This is the first time in which a nonmonotone technique is applied to nonsmooth equations. They claim that the monotone requirement can be even more harmful for systems of nonlinear equations. In fact in this case there is no longer a strong connection between the merit function and the original problem. For this reason, a good algorithm should use the structure of the original problem as much as possible (e.g., use the unit step size along the original Newton-type direction). Assumptions are given on general merit and auxiliary functions.

- Liu et al. [75]: the original nonmonotone technique (2.2) is applied on the top of a new theoretical framework that encapsulates many different line searches: exact, Curry-Altman's, Armijo's, Goldstein's, Wolfe's, De Leone-Grippo-Lucidi's. They employ a L-BFGS method.

- Kanzow [63]: the original nonmonotone technique (2.2) is applied on the squared norm merit function of linear complementarity problems. The algorithm is called noninterior continuation method. An interesting aspect here is the fact that the nonmonotone line search perform better than its monotone counterpart, even if the merit function is not too nonlinear.

- De Luca et al. [27]: the original nonmonotone technique (2.2) is applied on the semismooth reformulation of nonlinear complementarity problems. In those problems nonmonotone techniques are very well suited because the monotone line search often leads to very small step sizes.

- Toint [90]: a new nonmonotone technique (2.7) is applied to truncated Newton method. In particular he proposes to control the amount of nonmonotonicity by a measure indicating how well adapted the truncated Newton step is to the true objective function. This adequacy is measured by examining how close are the past step sizes to 1. Thanks to the numerical results assessed in [90] various observations are presented:

  - it is often advantageous to enforce monotonicity in the first few iterations;

  - the differences in CPU time between monotone and nonmonotone methods are most important for large problems;

  - when function evaluations is considered the gain provided by some of the analyzed nonmonotone methods is consistent, regardless of problem size;

  - in the absence of preconditioning, the new proposed variant appears to yield the highest potential for substantial efficiency gains while best limiting the probability of not solving problems.

$$
\begin{aligned}
f(x^k + \alpha^k d^k) \leq f(x^k) + \left(\prod_{j=1}^{j^k} \alpha^{k(l-j)} \right)^{\frac{1}{j^k}} \cdot (f^k_{max} - f(x^k)) \\
+ \gamma \alpha^k \nabla f(x^k)^T d^k, \\
\text{where } j^k := \arg \max_{0 \leq j \leq \min(k,W)} f(x^{k-j}).
\end{aligned}
\tag{2.7}
$$

- Ferris et al. [41]: they use the stabilization strategy from [51] for a negative curvature direction together with a curvilinear line search. Thanks to this strategy they are able to employ a high amount on nonmonotonicity.

  $$
  \begin{aligned}
  & f(x^k + {\alpha^k}^2 s^k + \alpha^k d^k) \leq \\
  & f^k_{max} + \gamma {\alpha^k}^2 \left(\nabla f(x^k)^T s^k + \frac{1}{2} {d^k}^T \nabla^2 f(x^k) d^k \right),
  \end{aligned}
  \tag{2.8}
  $$

  where s^k is a Newton-type direction, d^k is a negative curvature direction and $\nabla^2 f$ is the Hessian of f.

- Toint [91]: he proposes a new nonmonotone strategy (2.9) for trust-region methods. In this new technique, coherence on the nonmonotone trust-region approach is enforced by comparing both the achieved and predicted reductions for the last j^k iterations. Moreover, he investigates the importance of the nonmonotone window W, finding out that in some cases an infinite W might be the best choice. For this reason he also proposes a new strategy that should fix W in an adaptive way. This idea of adapting W has been very popular in many papers that followed [91]. In the first experiments he uses a delay on the nonmonotone effect, but keeping track of f^k_{max} from the beginning.

  $$
  \max \left\{ \frac{Ared^k}{Pred^k}; \frac{f^k_{max} - f(x^k + d^k)}{\sum_{i=0}^{j^k} Pred^{k-i}} \right\} \geq \gamma. \tag{2.9}
  $$

- Raydan [82]: the original nonmonotone line search (2.2) is employed to globalize the Barzilai-Borwein method. In fact for this method it is only possible to obtain global convergence in the two dimensional quadratic case, while it needs a globalization technique to be globally convergent in the general case. Barzilai-Borwein usually is highly nonmonotone and forcing monotonicity would reduce the method to a steepest descent. This is one of the most successful applications of nonmonotone strategies. From the numerical experiments, he observes that in general the choice of W does not have a great influence on results. The opposite is instead happening on problems with singular or very ill-conditioned Hessian at the solution. He claims that in these cases the choice of the parameter W is a delicate issue.

- Facchinei and Kanzow [33]: a Levenberg-Marquardt method for a semismooth equation reformulation of nonlinear complementarity problems. Global convergence is ensured by a nonmonotone line search (2.2) in which the reference iteration has been adapted in a similar way to the one proposed in [91]: keep W fixed as long as the algorithm seems to make progress and modify it only if for a certain prefixed number of consecutive steps the function values increase.

- Zhang and Chen [103]: in this paper the nonmonotone technique (2.2) is applied to globalize a Levenberg-Marquardt method for nonlinear least squares problems. A second nonmonotone strategy is also developed to obtain a higher degree of nonmonotonicity. In particular, they use the classical diagonal matrix employed to better invert the Levenberg-Marquardt equation, also in the quadratic approximation of f and thus in the predicted reduction.

- Lucidi et al. [77]: this paper extends the approach proposed in [41] to large-scale optimization problems. In particular, they show that it is possible to weaken the assumptions on the negative curvature directions to prove same convergence properties of the general stabilization framework from [41]. The curvilinear line search used is (2.8).

- Li and Fukushima [70]: a L-BFGS method that approximate Gauss-Newton direction is applied to solve symmetric nonlinear equations. A new line search (2.10) is applied to the merit function of the nonlinear equation and the resulting method possess only approximate norm descent property. This is the first time in which a user-defined $\{\epsilon^k\}$ sequence is exploited in a line search. In this case ϵ^k is not introduced to avoid the monotonic decrease of f, but instead to obtain the approximate norm descent property. Authors never cite [55] nor any other nonmonotone works.

$$\begin{aligned} f(x^k + \alpha^k d^k) \leq & f(x^k) - 2\sigma_1 \|\alpha^k d^k\|^2 - \sigma_2 \alpha^k f(x^k) + \epsilon^k f(x^k), \\ & \sum_{k=0}^{\infty} \epsilon^k < \epsilon < \infty. \end{aligned} \tag{2.10}$$

where $\sigma_1 > 0$, $\sigma_2 > 0$ and f is the merit function of the nonlinear equation.

- Birgin et al. [11]: this work is the extension of [82] to the convex constrained case. In fact, instead of the spectral gradient (Barzilai-Borwein method) a projected spectral gradient direction is employed. Note that the method appears to be a generalized steepest descent method, but it is clear from its derivation that it is related to the quasi-Newton family of methods through an approximated secant equation. This paper is another very successful application of the nonmonotone technique (2.2).

- Li and Fukushima [71]: this work is the extension to [70] in which the new Derivative Free (DF) line search (2.11) is used to achieve a norm descent property. In particular the term $\epsilon^k f(x^k)$ is introduced to to obtain a well-defined line search.

$$\begin{aligned} f(x^k + \alpha^k d^k) \leq & f(x^k) - 2\sigma_1 \|\alpha^k d^k\|^2 + \epsilon^k f(x^k), \\ & \sum_{k=0}^{\infty} \epsilon^k < \epsilon < \infty. \end{aligned} \tag{2.11}$$

- Dai and Zhang [24]: starting from [82] it was pointed out the importance of the choice of W. Thus in [24] they propose an adaptive way to update W. They take inspiration from [91] and his adaptive choice, but their rule is more elaborated.

- Ulbrich [94]: a new nonmonotone trust-region condition (2.12) is developed for box-constrained semismooth system of equations. He claims that nonmonotonicity helps preventing convergence to local nonglobal solutions, especially in the case of problems with a least-squares-like objective function. The new condition (2.12) requires that the f in new point is either less than $f(x^k)$ or less than a weighted mean of last function values (note that scalars are chosen by the user).

$$\frac{\max\left\{f(x^k), \sum_{i=0}^{j^k} \lambda_i f(x^{k-i})\right\} - f(x^k + d^k)}{\sum_{i=0}^{j^k} Pred^{k-i}} \geq \gamma, \quad \sum_{i=0}^{j^k} \lambda_i = 1 \quad (2.12)$$

- Zhang and Zhang [101]: in this paper on the sequential quadratic programming is applied the original nonmonotone technique (2.2). In particular, they apply it to a curvilinear line search exploiting a very weak degree of nonmonotonicity ($W = 1$).

- Dai [19]: various global and local convergence results are assessed in the case of unconstrained optimization and unspecified direction. He suggests to use a different nonmonotone strategy (2.13). In particular, whenever the initial step size has good properties it should be accepted as often as possible, but whenever (2.2) is not satisfied, he claims that there is no reason to obtain α^k by employing (2.2) and not (2.1). He states that when the function is not highly nonlinear one could prefer the monotone condition. He shows the good performance of the modified strategy with respect to the original (2.2) on some of the problems.

$$\begin{cases} (2.2) & \text{if } \alpha_0^k \text{ is accepted,} \\ (2.1) & \text{otherwise.} \end{cases} \quad (2.13)$$

- Dai et al. [23]: they propose two new initial step sizes for the algorithm from [82]. Also here the original nonmonotone line search (2.2) is used to globalize the method. Note that the nonmonotone line search is not exactly quadratic, since only the norm of the step is squared, while not the step size.

- Grippo and Sciandrone [53]: a new nonmonotone globalization scheme is developed for Barzilai-Borwein method. In particular, it is defined a new class of methods that employ watchdog techniques together with the line search (2.14). The unmodified Barzilai-Borwein is applied locally and the actual reduction obtained is checked with respect to some reference value. When the step is rejected, the method backtracks to a previous iterate and from there the nonmonotone line search technique (2.14) is employed.

$$f(x^k + \alpha^k d^k) \leq f_{max}^k + \gamma_1 \alpha^k \nabla f(x^k)^T d^k - \gamma_2(\alpha^k \|d^k\|)^2, \quad (2.14)$$

where $0 \leq \gamma_1 < 1$, $0 \leq \gamma_2$ and $0 < \gamma_1 + \gamma_2$.

- Sun et al. [88]: a new nonmonotone generalization framework is developed. Armijo, Goldstein and Wolfe line searches are showed to belong this class of line searches (2.15).

  $$f(x^k + \alpha^k d^k) \leq f_{max}^k - \sigma(t^k), \tag{2.15}$$

  where σ is a forcing function and $t^k = -\frac{\nabla f(x^k)^T d^k}{\|d^k\|}$.

- Facchinei et al. [34]: they use the stabilization technique from [51] to globalize a truncated Newton method for large-scale bound constrained problems. Thanks to the proposed algorithm is possible to obtain a sequence of unfeasible points. The merit function on which the nonmonotone line search is applied also takes into account the feasibility of the solution.

- Yu and Gao [96]: as in [105] nonmonotone line search (2.6) is applied to globalize sequential quadratic programming on minmax problems. This paper is extending [105] to the constrained case thanks to the use of a merit function.

- Plagianakos and Magoulas [80]: a new nonmonotone line search is applied to the batch training of multi-layer perceptron. The original nonmonotone technique (2.2) is applied together with a strategy to adapt the nonmonotone window W^k (2.16) on the base of a local approximation of the Lipschitz constant L^k. Experiments show that nonmonotonicity is valuable in the training of neural networks because, very often, the method escapes from local minima and flat valleys, whereas other methods are trapped.

  $$W^k = \begin{cases} W^{k-1} + 1 & L^k < L^{k-1} < L^{k-2}, \\ W^{k-1} - 1 & L^k > L^{k-1} > L^{k-2}, \\ W^{k-1} & \text{otherwise.} \end{cases} \tag{2.16}$$

- Birgin et al. [12]: they extend [11] to an inexact-projection version of the method, to be applied when the feasible set is not an easy set. The original (2.2) is employed also here.

- La Cruz and Raydan [68]: in this paper the nonmonotone spectral gradient from [82] is extended for solving large-scale nonlinear equations. They are the first to use the directions $d^k = \pm F(x^k)$ in a systematic way, where $F(x) = 0$ is the system of nonlinear equations. As in [82], Barzilai-Borwein initial step size is combined with the original nonmonotone line search (2.2). The resulting algorithm has been called Spectral Algorithm for Nonlinear Equations (SANE).

- Zhang and Zhang [102]: for the first time a nonmonotone technique is applied to an adaptive trust-region method. These methods are characterized by the fact that the trust-region radius depends on the gradient of the function in x^k. To the best of our knowledge this is the first time that the nonmonotone technique (2.17) is employed. Nonmonotone adaptive trust-region method will become very popular in papers that followed [102].

$$\frac{f^k_{max} - f(x^k + d^k)}{Pred^k} \geq \gamma. \tag{2.17}$$

- Zhang and Hager [100]: a new nonmonotone technique (2.18) is proposed to solve some potential drawbacks of the original (2.2):

  - the use of the max function might discard a good reference value $f(x^k)$;
  - in some cases numerical results are highly dependent on the choice of W;
  - as pointed out in [19], for R-linearly convergent sequence obtained when minimizing uniformly convex functions, it is in general not possible to prove that (2.2) is satisfied for any bounded fixed W. Note that in the case of Q-linearly convergent subsequence this is instead possible.

  The new proposed technique (2.18) does no more require a decrease w.r.t. f^k_{max}, but instead w.r.t. a convex combination of all the past function values. The parameter η^k controls how to combine these function values. If $\eta^k = 1$ the convex combination boils down to a simple mean of all the past function values. The new nonmonotone technique (2.18) has become very popular in papers that followed [100].

Thanks to (2.17) they are able to avoid the third drawback described above, even it is hard to consider it a real issue for a successful application of a nonmonotone technique. On the other hand, ignoring a good function value $f(x^k)$ might be considered a proper weakness of (2.2) and the new technique (2.18) actually fixes this issue when $f(x^k)$ has enough weight in the convex combination. In fact as W, also η^k is an important parameter to be chosen, even if η^k has probably less influence on the overall outcome. On the other side, W and more so η^k, have high impact on the results only on some particular problems, for most of the application both are very indirect parameter to control nonmonotonicity (see Chapter 5 for an accurate discussion on the topic).

$$\begin{gathered} f(x^k + \alpha^k d^k) \leq C^k + \delta \alpha^k \nabla f(x^k)^T d^k \\ Q^{k+1} = \eta^k Q^k + 1, \quad C^{k+1} = \frac{\eta^k Q^k C^k + f(x^k)}{Q^{k+1}}, \\ C^0 = f(x^0),\ Q^0 = 1. \end{gathered} \tag{2.18}$$

- Sun [87]: the technique (2.17) was been already developed in [102], even if he claims to be the first to employ it as a nonmonotone condition. Theoretical results are given in a different way, since the focus in [87] is the relationship between the trust-region method and line search approach.

- Dai and Fletcher [20]: they show that even nonmonotone line searches (in particular (2.2)) may significantly degrade the performances of the projected Barzilai-Borwein method. Moreover they show that even with a much larger W, performances are still degraded. As observed in [43], Barzilai-Borwein method is in fact very nonmonotone even at the beginning of the optimization procedure. This issue cannot be solved by increasing W, since by using (2.2) it is not possible to accept an iterate x^k for which we have $f(x^k) > f(x^0)$ (see Chapter 5). On the other side, they also show an example in which the original method without line search may not converge: it cycles between several points. For this reason they suggest to include a line search that would not degrade the performance of the unmodified method. In particular, they define a new strategy to adapt the reference iteration, in a similar way to that used in [91], and that allows $f(x^k) > f(x^0)$ on some iterates. Box constrained quadratic programming is assessed in this paper.

- Fu and Sun [44]: as in [102] they focus on unconstrained optimization problems applying a nonmonotone adaptive trust-region method. Differently from [102] they employ (2.19) instead of (2.17) and they approximately solve the trust-region subproblem in a different way.

$$\frac{f^k_{max} - f(x^k + d^k)}{f^k_{max} - \Phi^k(0) - \Phi^k(d^k)} \geq \gamma. \tag{2.19}$$

- Zanni [99]: to solve constrained quadratic programming and the training of SVM as a particular case, he employs a decomposition approach and a spectral projected gradient in each subproblems. To the best of our knowledge it is the first time in which a nonmonotone technique is combined with a decomposition method. In particular he employs (2.2) together with the adaptive reference update by [20].

- Hager and Zhang [58]: as often happens in presence of Barzilai-Borwein methods, also in this work a high amount of nonmonotonicity is needed in the early stage of the optimization procedure. Condition (2.2) is in fact too strict because it enforces $f(x^k) > f(x^0)$. Convergence proof is obtained for general active set algorithms combined with line searches that can ensure that the function value in the reference iteration is infinitely often less or equal than f_{max}. In the appendix they define a new nonmonotone strategy for an adaptive reference iteration based on the minimum f found and the amount of iterations in which the initial step size α^k_0 is first accepted.

- La Cruz et al. [67]: in this paper a DF extension of SANE [68] is developed. A new nonmonotone strategy (2.20) that sum together the original f^k_{max} and ϵ^k from (2.10) is developed. Weak global convergence is obtained by additionally assuming that limit points for which we have $\nabla f(x^k)^T F(x^k) = 0$ are not solutions of the nonlinear equations. Also in this application of a spectral method a high amount of nonmonotonicity is very helpful in all the stages of the optimization procedure.

$$\begin{aligned} f(x^k + \alpha^k d^k) \leq & f^k_{max} + \epsilon^k + \gamma\alpha^k \nabla f(x^k)^T d^k, \\ & \sum_{k=0}^{\infty} \epsilon^k < \epsilon < \infty. \end{aligned} \tag{2.20}$$

- Dai et al. [22]: they propose an adaptive way to decide the length of the cycle of cyclic Barzilai-Borwein. The nonmonotone strategy is the same from [24], which is also adaptively modifying the reference iteration.

- Dai and Fletcher [21]: based on [20] they propose a new nonmonotone strategy for adapting the reference iteration. In addition they use an exact line search whenever the nonmonotone condition is not satisfied. Single equality box-constrained quadratic programming are considered in this paper.

$$\begin{cases} (2.2) & \text{if } \alpha_0^k \text{ is accepted,} \\ \text{Exact} & \text{otherwise.} \end{cases} \tag{2.21}$$

- Shi and Shen [85]: they propose a new nonmonotone line search (2.22) that sums the classical f_{max} to a quadratic term. Since (2.22) is now highly nonmonotone they use a very small nonmonotone window $W = 3$.

$$f(x^k + \alpha^k d^k) \leq f_{max}^k + \gamma \alpha^k \nabla f(x^k)^T d^k + \frac{1}{2} {\alpha^k}^2 {d^k}^T B^k d^k, \tag{2.22}$$

where B^k is an approximation of the Hessian matrix.

- Sun and Zhou [89]: in this paper they generalize the second order Goldstein line search to the nonmonotone case (2.23). In particular (s^k, d^k) is a descent pair of direction, e.g., a Newton and a negative curvature direction.

$$\begin{aligned} & f(x^k + {\alpha^k}^2 s^k + \alpha^k d^k) \leq \\ & f_{max}^k + \gamma_1 {\alpha^k}^2 \left(\nabla f(x^k)^T s^k + \frac{1}{2} {d^k}^T \nabla^2 f(x^k) d^k \right), \\ & f(x^k + {\alpha^k}^2 s^k + \alpha^k d^k) \geq \\ & f(x^k) + \gamma_2 {\alpha^k}^2 \left(\nabla f(x^k)^T s^k + \frac{1}{2} {d^k}^T \nabla^2 f(x^k) d^k \right). \end{aligned} \tag{2.23}$$

- Mo et al. [78]: they propose a new nonmonotone technique for trust-region (2.24). They adapt (2.18), originally proposed for line search, to the trust-region globalization technique.

$$\frac{C^k - f(x^k + d^k)}{\Phi^k(0) - \Phi^k(d^k)} \geq \gamma \quad \text{with } C^k \text{ from (2.18).} \tag{2.24}$$

- Serafini et al. [83]: they use a nonmonotone spectral projected gradient method for solving constrained quadratic programming and training of SVMs as a special case. In particular (2.2) is the nonmonotone line search employed. Moreover they develop a method for alternating between two different spectral step sizes.

- Gu and Mo [57]: a new nonmonotone technique for trust-region is proposed. In (2.25) there is a simple combination between $f(x^k)$ and the previous D^k. If a trial step is not accepted, the algorithm performs the nonmonotone line search to find an iterative point instead of solving the trust-region subproblem. Both trust-region and line search are nonmonotone. In the numerical results they modify η^k dynamically. In particular, they use values closer to 1 (more nonmonotonicity) when iterates are far away from the optimum and closer to 0 (less nonmonotonicity) when iterates approach the solution.

$$\begin{cases} \frac{D^k - f(x^k+d^k)}{\Phi^k(0)-\Phi^k(d^k)} \geq \gamma & \text{if } d_0^k \text{ is accepted,} \\ f(x^k+\alpha^k d^k) \leq D^k + \delta\alpha^k \nabla f(x^k)^T d^k & \text{otherwise.} \end{cases}$$
$$\text{where } \; D^k = \begin{cases} f(x^k) & k=1 \\ \eta^k D^{k-1} + (\eta^k - 1) f(x^k) & k>1 \end{cases} \tag{2.25}$$

- Yu and Pu [98]: in the framework proposed by [88] they show that using the nonmonotone technique (2.12) (adapted to the line search case) it is possible to obtain global convergence without the classical condition that relates the norm of the direction and the norm of the gradient ($\|d^k\| \leq c_1 \|\nabla f(x^k)\|$). It is interesting to point out that (2.12) is not as nonmonotone as (2.2).

- Shi and Guo [84]: they propose a new nonmonotone technique (2.26) that exploits L^k, an approximation of the Lipschitz constant. Various possible choices are given to approximate it.

$$f(x^k + \alpha^k d^k) \leq f_{max}^k + \gamma\alpha^k \left(\nabla f(x^k)^T d^k + \frac{1}{2}\alpha^k L^k \|d^k\|^2 \right). \tag{2.26}$$

- Su and Pu [86]: to solve equality constrained optimization problems the nonmonotone idea is also applied to the filter algorithm. Thus, a new step is either accepted when the objective or the violation function decrease. Exploiting (2.27), a modified version of (2.12), the filter condition needs to be satisfied only every W iteration.

$$\frac{\max\left\{f(x^k); \sum_{i=0}^{W} \lambda_i f(x^{k-i})\right\}}{Pred^k} \geq \gamma. \tag{2.27}$$

- Ahookhosh and Amini [1]: in this work an effective adaptive trust-region rule is combined with the simple nonmonotone technique (2.17), giving birth to an algorithm that received attentions from papers that followed.

- Hu et al. [61]: starting from the idea of [100], they propose a new nonmonotone technique (2.28) in which the convex combination of the function values is limited to a finite number of previous iterations.

$$\begin{aligned} f(x^k + \alpha^k d^k) &\leq C^k + \delta\alpha^k \nabla f(x^k)^T d^k, \\ Q^{k+1} &= 1 + \eta^k \sum_{i=0}^{W} \eta^{k-i}, \\ C^{k+1} &= \frac{\eta^k \sum_{i=0}^{W} \eta^{k-i} f(x^{k-i}) + f(x^k)}{Q^k} \end{aligned} \tag{2.28}$$

- Zhang and Huang [104]: an existing smoothing-type algorithm for solving systems of inequalities is extended to the use of the nonmonotone line search (2.18) and to the case of equalities together with inequalities.

- Ahookhosh et al. [2]: this paper is very similar to [57], both adaptive trust-region methods for unconstrained optimization, both use exactly (2.25). In the numerical results of [2] they also compare with [57], but they never explain the differences with their implementation. The claim that [100] proved that best convergence results are obtained by stronger nonmonotone strategy when iterations are far from the optimum, and by weaker nonmonotone strategy when iterations are close to it. This conclusion does not take into account statements in many other works that claim the exact opposite.

- Fletcher [42]: even if he does not make uses of any common globalization techniques, function values in each internal phase (a "sweep") can increase. He also claims that for some application Barzilai-Borwein method was both used with $W = 2, 3$ or $W = 20, 50$. He concludes that the best choice of W is problem dependent.

- Zhou [106]: this is a small note for the globalization of the PRP method using (2.10). In the paper it is only showed weak convergence to stationary points.

- Grippo and Rinaldi [52]: to the best of our knowledge this is the first nonmonotone theoretical framework for DF optimization. They first extend known methods like Hooke-Jeeves and Rosenbrock to the nonmonotone case, then they develop a new algorithm based on the rotation of the coordinate axes. Nonmonotone technique (2.2) is combined with a quadratic line search to avoid the use of the gradient.

- Li et al. [69]: for solving equality constrained minimization problems they combine for the first time an augmented Lagrangian multiplier method together with a nonmonotone line search (2.18). In this case, the nonmonotonicity is employed to quickly accept large steps on the "hard" direction. Note that f may not be differentiable with respect to one block of variables.

- Liu and Ma [76]: they introduce a new nonmonotone technique (2.29) that combines the ratio between actual and predicted reduction to the line search mapping as it has been done in (2.25). Instead of using only the first order information in the line search, they sum a quadratic term that additionally exploits an approximation of the Lipschitz constant L^k.

$$\begin{cases} \frac{D^k - f(x^k + d^k)}{\Phi^k(0) - \Phi^k(d^k)} \geq \gamma & \text{if } d_0^k \text{ is accepted,} \\ f(x^k + \alpha^k d^k) \leq D^k & \\ \qquad\qquad + \delta \alpha^k \nabla f(x^k)^T d^k + \frac{1}{2} \alpha^k L^k \|d^k\|^2 & \text{otherwise.} \end{cases}$$
$$\text{where } D^k = \begin{cases} f(x^k) & k = 1 \\ \eta^k D^{k-1} + (\eta^k - 1) f(x^k) & k > 1 \end{cases} \tag{2.29}$$

- Amini et al. [4]: they propose a new nonmonotone technique (2.30) that simply combines f^k_{max} and $f(x^k)$. The value η^k is adapted during the optimization process, using a higher amount of nonmonotonicity in the beginning and decreasing it in later iterations. Theory is developed without fully characterizing the direction, but in the experiment they use 3 different optimization methods: Barzilai-Borwein, L-BFGS and truncated Newton. They compare their new technique with [55, 100].

$$\begin{aligned} &f(x^k + \alpha^k d^k) \leq R^k + \gamma \alpha^k \nabla f(x^k)^T d^k, \\ &R^k := \eta^k f^k_{max} + (1 - \eta^k) f(x^k). \end{aligned} \tag{2.30}$$

- Ahookhosh et al. [3]: they propose two new nonmonotone techniques that compute a convex combination of function value, as in [100], but taking into account only last W values. The only difference between the two techniques is how to handle iterations k with $k < W$. They compare their two techniques with [2, 4, 55, 100].

$$\begin{aligned} &f(x^k + \alpha^k d^k) \leq T^k + \gamma \alpha^k \nabla f(x^k)^T d^k, \\ &T^k := (1 - \eta^{k-1}) f(x^k) + \eta^{k-1}(1 - \eta^{k-2}) f(x^{k-1}) + \dots \\ &\quad + \eta^{k-1} \cdot \ldots \cdot \eta^{k-W} f(x^{k-W}). \end{aligned} \tag{2.31}$$

- Nosratipour et al. [79]: they propose two new adaptive nonmonotone rules based on the morphology of the objective function. One of the techniques exploits (2.2) and adaptively updates W almost as in (2.16), but instead of using the approximated Lipschitz constant to control it, they use the norm of the gradient (2.32). The other technique computes a convex combination of function value as in [3], but the distribution of weights is obtained differently. They compare their two techniques with [4, 55, 80, 100].

$$W^k = \begin{cases} W^{k-1} + 1 & 10^{-1} < \|\nabla f(x^k)\|_\infty, \\ W^{k-1} & 10^{-3} \leq \|\nabla f(x^k)\|_\infty \leq 10^{-1}, \\ W^{k-1} - 1 & \text{otherwise.} \end{cases} \tag{2.32}$$

In Table 2.1, a summary of all the papers cited in this chapter is reported. For each of them we gave a fast reference to the method used, the nonmonotone technique employed and the application for which it has been designed.

Paper	Method	Application	Cond.
[55]	Newton + Armijo	Unconstrained	(2.2))
[50]	TN + Armijo	Unconstrained	(2.2)
[51]	Unspecified + Armijo	Unconstrained	(2.4)
[28]	TR	Unconstrained	(2.5)
[105]	SQP + Armijo/Arc	Minimax	(2.6)
[40]	Newton + Armijo	NLE	(2.2)
[75]	L-BFGS + General Framework	Unconstrained	(2.2)
[63]	Newton + Armijo	LCP	(2.2)
[27]	SS Newton+Armijo/Quadratic	NCP	(2.2)
[90]	CG + Armijo	Unconstrained	(2.7)
[41]	Negative Curvature+Curvilinear	Unconstrained	(2.8)
[91]	TR	Convex Set	(2.9)
[82]	BB + Armijo	Unconstrained	(2.2)
[33]	NS LM + Armijo/Quadratic	NCP	(2.2)
[103]	LM	Least Squares	(2.2)
[77]	Negative Curvature+Curvilinear	Unconstrained	(2.8)
[70]	Approximated GN + Quadratic	Symmetric NLE	(2.10)
[11]	SPG + Armijo	Convex Set	(2.2)
[71]	Broyden QN + Quadratic	NLE	(2.11)
[24]	BB + Armijo	Unconstrained	(2.13)
[94]	TR + Projected Newton	SS NLE+Box+Equality	(2.12)
[101]	SQP+Curvilinear	Constrained	(2.2)
[19]	Unspecified + Armijo	Unconstrained	(2.13)
[23]	BB + Quadratic	Unconstrained	(2.2)
[53]	BB Armijo+Quadratic	Unconstrained	(2.14)
[88]	General Framework	Unconstrained	(2.15)
[34]	Active-set + Armijo	Box	(2.2)
[96]	SQP+Armijo/Curvilinear	Minimax Constrained	(2.6)
[80]	BB + Armijo	Neural Nets	(2.2)
[12]	SPG + Armijo	Convex Set	(2.2)
[68]	SANE	NLE	(2.2)
[102]	Adaptive TR	Unconstrained	(2.17)
[100]	L-BFGS + Wolfe	Unconstrained	(2.18)
[87]	TR	Unconstrained	(2.17)
[20]	BB + Armijo	QP + Box	(2.2)
[44]	Adaptive TR	Unconstrained	(2.19)
[99]	Decomposition + Inner SPG	QP (SVM)	(2.2)
[58]	Projected Gradient + Armijo	Box	(2.2)
[22]	Cyclic BB + Armijo	Unconstrained	(2.13)

[21]	Secant Approximation + Exact	QP+Box+Equality	(2.21)
[85]	Specific	Unconstrained	(2.22)
[89]	Negative Curvature+Goldstein	Unconstrained	(2.23)
[78]	TR	Unconstrained	(2.22)
[83]	SPG + Armijo	QP (SVM)	(2.2)
[57]	TR + Specific	Unconstrained	(2.25)
[98]	Unspecified+(2.15)	Unconstrained	(2.12)
[84]	CG + Specific	Unconstrained	(2.26)
[86]	Filter + TR	Equality Constrained	(2.27)
[1]	Adaptive TR	Unconstrained	(2.17)
[61]	Unspecified + Wolfe	Unconstrained	(2.28))
[104]	Smoothing + Armijo	NLE + Inequalities	(2.18)
[2]	Adaptive TR + Armijo	Unconstrained	(2.25)
[42]	BB + Specific	Unconstrained	No
[69]	(AD + BB) + Armijo	NS + Equality	(2.18)
[76]		Unconstrained	(2.18)
[106]	PRP + Quadratic	Unconstrained	(2.10)
[52]	DF + Quadratic	Unconstrained	(2.2)
[4]	Unspecified + Armijo	Unconstrained	(2.30)
[3]	Unspecified + Armijo	Unconstrained	(2.31)
[79]	Unspecified + Armijo	Unconstrained	(2.32)

Table 2.1: A summary of all the papers cited in this chapter, giving a fast reference to the method used, the nonmonotone technique employed and the application for which it has been designed.

Chapter 3

A Nonmonotone Trust-Region Method for Generalized Nash Equilibrium Problems

Generalized Nash equilibrium problems (GNEPs) are often difficult to solve by Newton-type methods since these problems tend to have locally nonunique solutions. Here we take an existing trust-region method which is known to be locally fast convergent under an error bound condition, and modify this method by a nonmonotone strategy in order to obtain a more reliable and efficient solver. The nonmonotone trust-region method inherits the nice local convergence properties of its monotone counterpart and is also shown to have the same global convergence properties. Numerical results indicate that the nonmonotone trust-region method is significantly better than the monotone version, and is at least competitive to an existing software applied to the same reformulation used within our trust-region framework. Additional tests on Quasi-Variational Inequalities (QVI) are also presented to validate the efficiency of the proposed extension.[1]

[1] This chapter has been published as "A Nonmonotone Trust-Region Method for Generalized Nash Equilibrium and Related Problems with Strong Convergence Properties" in *Computational Optimization and Application vol. 69, iss. 3, pp. 629-652, 2018* [46].

3.1 Introduction

The aim of this chapter is to construct an efficient method for the solution of generalized Nash equilibrium problems (GNEPs for short). These problems have a wide range of applications in economics, operations research, computer science, telecommunications etc. The interested reader is referred to the survey papers [35, 38] for more details regarding applications, theoretical results and numerical approaches for the solution of GNEPs.

The main difficulty with GNEPs is that these problems tend to have solution sets which are even locally nonunique. This means that standard Newton-type schemes typically do not work very well. This is also illustrated, for example, in the more general context of quasi-variational inequality problems by the semismooth Newton method from [36] which works extremely well for some examples, but fails for quite a few other test problems. On the other hand, there exist some very reliable methods with nice global convergence properties like the interior-point-type scheme from [32] or the augmented Lagrangian-type method from [64], but they are not locally fast convergent, and might even have problems in getting high accuracy of the solutions.

In order to obtain a globally and locally superlinearly convergent method for GNEPs, we are therefore urged to apply suitable methods which also work for nonunique solutions. Fortunately, in the meantime, there exist a few methods for optimization problems and nonlinear systems of equations which have this desired property under an error bound condition, see, e.g., [25, 35, 39, 65, 95] for some attempts in this direction. Moreover, there also exist some recent papers that provide error bounds for GNEPs [31, 62]. However, these error bounds depend on the particular reformulation of the GNEP. The most prominent reformulations take the KKT conditions of the players, concatenate all KKT conditions into a larger system and apply, e.g., the Fischer-Burmeister function in order to get a semismooth system of equations. Unfortunately, it turns out to be difficult to find Newton-type methods for nonsmooth systems of equations which converge locally superlinearly under an error bound condition without any further assumptions, see, for example, the discussion in [59].

We therefore use a smooth reformulation of the GNEP with some simple bound constraints for which a suitable error bound is available from [31]. Moreover, we take the trust-region method from [92] which works precisely in our situation where we have a smooth constrained system of equations and which is locally superlinearly convergent under an error bound condition. In order to improve the practical convergence of the trust-region method from [92], we introduce a nonmonotone variant of that method in such a way that it is still globally convergent and inherits the local properties of the original method.

The chapter is therefore organized in the following way: Section 3.2 presents our algorithmic scheme. There we first recall the (monotone) trust-region method from [92] and then derive the necessary modifications for a nonmonotone version with the same local convergence properties. The global convergence of the nonmonotone trust-region method is shown in Section 3.3. The details for the application of this method applied to GNEPs are presented in Section 3.4. The corresponding numerical results are given in Section 3.5, together with some additional tests on Quasi-Variational Inequalities (QVIs for short). We then conclude with some final remarks in Section 3.6.

Notation: $\mathbb{R}^n$ denotes the n-dimensional Euclidean vector space, $\mathbb{R}^n_+$ is its subset in which vectors have only nonnegative components, the symbol $\|\cdot\|$ is the Euclidean vector norm, G' is the Jacobian of a generic differentiable mapping G, ∇G is its transposed and $\nabla_{z_\nu} G$ is the same, but with respect to the variables z_ν, where z_ν typically indicates a suitable subvector of z. The symbol $B_r(z)$ denotes the (Euclidean) ball of radius $r > 0$ around a given point z, while $P_\Omega(z)$ stands for the (Euclidean) projection of z onto a nonempty, closed and convex set Ω. Iterates of the algorithms are denoted by x^k.

3.2 Trust-Region Methods

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a given function and $\Omega \subseteq \mathbb{R}^n$ be a nonempty set. We consider the problem of finding a solution of the constrained nonlinear system of equations

$$F(x) = 0, \quad x \in \Omega \tag{3.1}$$

which is of much interest for its own since problems of this kind arise in many situations. The GNEP discussed in some more detail in Section 3.4 is only one of the possible applications for which the original problem might be reduced to solve 3.1. The solution set of (3.1) will be denoted by X^*.

3.2.1 The Monotone Trust-Region Method

In this section we report the trust-region method by Tong and Qi [92], formulated in a way such that its generalization to a nonmonotone framework will be easy to state with just some minor modifications.

We first recall or state the assumptions that are assumed to hold for problem (3.1) in order to get global and local fast convergence of the trust-region method.

Assumption 1. **(a)** F is continuously differentiable with F' being locally Lipschitzian.

(b) Ω is nonempty, closed, and convex.

(c) The solution set X^* is nonempty.

(d) $\|F(x)\|$ provides a local error bound in a neighborhood of a solution $x^* \in X^*$, i.e. there exists constants $\delta > 0$ and $\gamma > 0$ such that

$$\operatorname{dist}(x, X^*) \leq \gamma \|F(x)\| \quad \forall x \in \Omega \cap B_\delta(x^*).$$

Note that the central condition in Assumption 1 is part (d) where the usual nonsingularity condition of $F'(x^*)$ is replaced by a (weaker) error bound. This condition is only required in the local analysis in order to prove local fast convergence properties.

In order to describe the trust-region method, let us introduce the merit function

$$\Psi(x) := \frac{1}{2}\|F(x)\|^2$$

associated to (3.1). Since X^* is nonempty, x^* solves (3.1) if and only if x^* is a solution of the optimization problem

$$\min_x \ \Psi(x) \quad \text{s.t.} \quad x \in \Omega. \tag{3.2}$$

We call x^* a *stationary point* of (3.1) if x^* is a stationary point of the corresponding optimization problem (3.2), i.e. if

$$\nabla\Psi(x^*)^T(x - x^*) \geq 0 \quad \forall x \in \Omega;$$

recall that this is equivalent to x^* satisfying the fixed-point equation

$$x^* = P_\Omega\big(x^* - \gamma\nabla\Psi(x^*)\big)$$

for an arbitrary constant $\gamma > 0$.

To deal with locally nonunique solutions, the trust-region method considers a subproblem where the objective function involves an additional regularization term. More precisely, the regularized trust-region subproblem at a current iterate x^k is given by

$$\begin{aligned} &\min\ \Phi^k(d) := \frac{1}{2}\|F(x^k) + F'(x^k)d\|^2 + \frac{1}{2}\mu^k\|d\|^2 \\ &\text{s.t. } \|d\| \leq \Delta, \end{aligned} \tag{3.3}$$

where $\Delta > 0$ denotes the trust-region radius and $\mu^k > 0$ is a suitable constant depending on the iteration index k. This notation allows us to state the following trust-region method which corresponds to the method from [92], except that we leave the choice of an appropriate scalar $\hat{r}^k$ unspecified for the moment.

Let us give a few comments to explain Algorithm 1: in our convergence analysis we always assume implicitly that the termination criterion in Step 1 does not hold after finitely many iterations. Hence we assume in our theoretical analysis of Algorithm 1 that none of the iterates x^k is an exact stationary point of problem (3.2), so that $P_\Omega\big[x^k - \nabla\Psi(x^k)\big] - x^k \neq 0$ or, equivalently, that $\overline{d}_G^k(\Delta) \neq 0$ holds for all k. The remainder of Step 1 initializes some parameters for the inner iteration starting in Step 2 In particular, we reset the trust-region radius Δ^k by taking its projection onto the interval $[\Delta_{\min}, \Delta_{\max}]$.

Step 2 then computes a projected gradient direction which, more or less, is responsible for the global convergence of Algorithm 1. The projected trust-region step from Step 3, on the other hand, is the main ingredient to verify local fast convergence under an error bound assumption. In Step 4, a convex combination of these two directions is taken which yields the smallest objective function of the standard quadratic approximation of the mapping Ψ. Note that the optimal $t^*(\Delta)$ in this step can be computed analytically. In fact, it is not difficult to see that it has the closed form expression

$$t^*(\Delta) = \max\big\{0, \min\{1, t(\Delta)\}\big\},$$

Algorithm 1: Trust-Region Framework

```
Input: $x^0 \in \Omega$ starting point and constants $0 < \alpha_1 < 1 < \alpha_2,\ 0 < \rho_1 < \rho_2 < 1,\ \eta \in (0,1),\ \sigma \in (0,1),\ C > 0,\ \Delta_0 > 0,\ \Delta_{max} > \Delta_{min} > 0$. Set $k := 0$.
1 If $x^k$ is a stationary point of the optimization problem (3.2): STOP. Otherwise
    set $\Delta^k := \min\{\Delta_{max}, \max\{\Delta_{min}, \Delta^k\}\}$, $\Delta := \Delta^k$, $\mu^k := C\|F(x^k)\|$ .
2 Projected Gradient Direction: Compute
      $d_G^k(\Delta) := -\Big(\frac{\Delta}{\Delta_{max}}\Big)\gamma^k \nabla\Psi(x^k),$   (3.4)
      $\bar{d}_G^k(\Delta) := P_\Omega\big[x^k + d_G^k(\Delta)\big] - x^k,$   (3.5)
    with $\gamma^k := \min\Big\{1, \frac{\Delta_{max}}{\|\nabla\Psi(x^k)\|}, \frac{\eta\Psi(x^k)}{\|\nabla\Psi(x^k)\|^2}\Big\}$.
3 Projected Trust-Region Direction: Solve the trust-region subproblem (3.3) by a
    suitable algorithm and denote its solution by $d_{tr}^k(\Delta)$. Then compute
      $\bar{d}_{tr}^k(\Delta) := P_\Omega[x^k + d_{tr}^k(\Delta)] - x^k.$
4 Optimal Combined Direction: Compute
      $\bar{d}^k(\Delta) := t^*(\Delta)\bar{d}_G^k(\Delta) + \big(1 - t^*(\Delta)\big)\bar{d}_{tr}^k(\Delta),$
    where $t^*(\Delta) \in [0,1]$ is an optimal solution of the problem
      $\min_{t\in[0,1]} q_\Delta^k(t) := \frac{1}{2}\big\|F(x^k) + F'(x^k)\Big(t\bar{d}_G^k(\Delta) + (1-t)\bar{d}_{tr}^k(\Delta)\Big)\big\|^2,$
5 Updates: Define the actual and predicted reductions by
      $Ared^k(\Delta) := \Psi\big(x^k + \bar{d}^k(\Delta)\big) - \Psi(x^k),$
      $Pred^k(\Delta) := \frac{1}{2}\big\|F(x^k) + F'(x^k)\bar{d}^k(\Delta)\big\|^2 - \Psi(x^k),$
    respectively, and choose a suitable scalar $\hat{r}^k$. If the following two conditions
      $-Pred^k(\Delta) \geq -\sigma\nabla\Psi(x^k)^T\bar{d}_G^k(\Delta)$   (3.6)
      $\hat{r}^k \geq \rho_1$   (3.7)
    hold, set
      $x^{k+1} := x^k + \bar{d}^k(\Delta), \quad \Delta_{k+1} := \begin{cases}\Delta & \text{if } \rho_1 \leq \hat{r}^k < \rho_2,\\ \alpha_2\Delta & \text{if } \hat{r}^k \geq \rho_2,\end{cases}$
    define ${\Delta^k}^* := \Delta$ as the trust-region radius that allowed $x^k + \bar{d}^k(\Delta)$ to be
    accepted, set $k \leftarrow k + 1$, and go to Step 1. Otherwise set $\Delta := \alpha_1\Delta$, and go to
    Step 2.
```

where $t(\Delta)$ is the solution of $\nabla q_\Delta^k(t) = 0$ and is given by

$$t(\Delta) = \begin{cases} \frac{-\left(F(x^k)+F'(x^k)\bar{d}^k_{tr}(\Delta)\right)^T F'(x^k)\left(\bar{d}^k_G(\Delta)-\bar{d}^k_{tr}(\Delta)\right)}{\|F'(x^k)\left(\bar{d}^k_G(\Delta)-\bar{d}^k_{tr}(\Delta)\right)\|^2} \\ \qquad\qquad\qquad\qquad \text{if } F'(x^k)\bar{d}^k_G(\Delta) \neq F'(x^k)\bar{d}^k_{tr}(\Delta), \\ \text{any number in } (-\infty, +\infty) \\ \qquad\qquad\qquad\qquad \text{if } F'(x^k)\bar{d}^k_G(\Delta) = F'(x^k)\bar{d}^k_{tr}(\Delta). \end{cases}$$

In Step 5, we first compute the actual and predicted reductions at the current point. The subsequent update rule in Step 5 depends on the suitable choice of $\hat{r}^k$. The standard choice corresponding to the monotone trust-region method from [92] is

$$\hat{r}^k := \frac{Ared^k(\Delta)}{Pred^k(\Delta)} \tag{3.8}$$

and completely specifies Algorithm 1. The updates in Step 5 are then similar to a standard trust-region update except that the new point is accepted only if the two conditions (3.6) and (3.7) hold. This second condition, which is standard for monotone trust-region method, will be relaxed in our nonmonotone version. The reason for having the additional criterion from (3.6) comes from the fact that we need to have the predicted reduction to be a negative number whenever we leave the inner iteration. This property is not clear a priori, but will be a consequence of Lemma 1 below.

Note that, by construction, all iterates x^k generated by Algorithm 1 belong to the feasible set Ω. Furthermore, let us recall from [92] that Algorithm 1 with the standard choice of $\hat{r}^k$ from (3.8) is

- well-defined, in particular, for each outer iteration k, the number of inner iterations between Step 2 and 5 is finite,
- globally convergent in the sense that every accumulation point is a stationary point of (3.2), and
- locally fast convergent under the relatively weak error bound condition from Assumption 1.

For the details, we refer the reader to [81, 92].

We close this section by stating two results that were given for a somewhat different active-set-type trust-region method in [81], but whose statements also hold for the above (monotone) trust-region method. These results will be used in our subsequent convergence analysis. For the sake of completeness, we provide the full proofs in an appendix. The first of these results corresponds to [81, Lemma 4.3].

Lemma 1. Consider the trust-region method from Algorithm 1 with the update $\hat{r}^k$ from (3.8). Then, for all $k \in \mathbb{N}$ and all $\Delta \in (0, \Delta_{max}]$, it holds that

$$\nabla \Psi(x^k)^T \bar{d}_G^k(\Delta) \leq -\left(\frac{\Delta}{\Delta_{max}\gamma^k}\right)\left\|\bar{d}_G^k(\Delta_{max})\right\|^2.$$

Note that this result implies that the predicted reduction is a negative number whenever we leave the inner iteration in Algorithm 1, i.e. we always have

$$Pred^k(\Delta^{k^*}) < 0, \tag{3.9}$$

cf. (3.6).

We next state a technical result which is the counterpart of [81, Proposition 4.1] and whose proof is also given in the appendix.

Proposition 1. Consider the monotone trust-region method from Algorithm 1 with the update $\hat{r}^k$ from (3.8). Suppose that x^* is an accumulation point of a subsequence $\{x^k\}_{k \in K}$. If x^* is not a stationary point, then there exist an index $\hat{k} > 0$ and a constant $\bar{\Delta} > 0$ such that, for all $k \in K$ with $k \geq \hat{k}$, (3.6) and (3.7) hold for all $\Delta \in (0, \bar{\Delta})$.

3.2.2 The Nonmonotone Trust-Region Method

Here we present a nonmonotone version of Algorithm 1. The main idea is to accept also suitable points which do not necessarily reduce the objective function value. This is achieved by accepting the new step $\bar{d}^k(\Delta)$ more frequently. To this end, we relax condition (3.7). The strategy is inspired by the work [91] by Toint. We will see in Section 3.5 that the nonmonotonicity improves the numerical behavior of the trust-region method.

To give a precise statement of the nonmonotone trust-region method, let us introduce one further parameter $W \in \mathbb{N}$. We then define

$$\Psi_{max}^k := \max_{k-W \leq i \leq k} \Psi(x^i) \tag{3.10}$$

as the largest function value among the last few iterations, where, formally, we set $x^{-1} := x^{-2} := \ldots := x^{-W} := x^0$ (or, alternatively, we can replace W by $W^k := \min\{k, W\}$). Since we are interested in the behavior of an infinite sequence, we may assume without loss of generality that we always have $k \geq W$. Furthermore, let us define

$$r(k) \leq k \text{ as the (say, largest) iteration index such that } \Psi(x^{r(k)}) = \Psi^k_{max} \text{ holds.} \tag{3.11}$$

We then define a modified (nonmonotone) actual reduction at iteration k by

$$Ared^{k\prime}(\Delta) := \Psi\big(x^k + \bar{d}^k(\Delta)\big) - \Psi^k_{max}. \tag{3.12}$$

Using an elementary calculation, we obtain the representation

$$\begin{aligned} Ared^{k\prime}(\Delta) &= \Psi\big(x^k + \bar{d}^k(\Delta)\big) - \Psi^k_{max} \\ &= \Psi\big(x^k + \bar{d}^k(\Delta)\big) - \Psi(x^{r(k)}) \\ &= \Psi\big(x^k + \bar{d}^k(\Delta)\big) - \Psi(x^k) + \sum_{i=r(k)}^{k-1} \big[\Psi(x^{i+1}) - \Psi(x_i)\big] \\ &= \Psi\big(x^k + \bar{d}^k(\Delta)\big) - \Psi(x^k) + \sum_{i=r(k)}^{k-1} \big[\Psi\big(x_i + \bar{d}_i(\Delta_i^*)\big) - \Psi(x_i)\big] \\ &= Ared^k(\Delta) + \sum_{i=r(k)}^{k-1} Ared_i(\Delta_i^*), \end{aligned}$$

where Δ_i^* is defined as in Step 5 of Algorithm 1. This representation motivates to define a corresponding modified (nonmonotone) predicted reduction by

$$Pred^{k\prime}(\Delta) := Pred^k(\Delta) + \sum_{i=r(k)}^{k-1} Pred_i(\Delta_i^*). \tag{3.13}$$

The idea is then to replace (3.7) by the condition

$$\max\left\{\frac{Ared^k(\Delta)}{Pred^k(\Delta)}, \frac{Ared^{k\prime}(\Delta)}{Pred^{k\prime}(\Delta)}\right\} \geq \rho_1. \tag{3.14}$$

Hence, the nonmonotone trust-region method is fully specified by Algorithm 1 with the choice

$$\hat{r}^k := \max\left\{\frac{Ared^k(\Delta)}{Pred^k(\Delta)}, \frac{Ared^{k\prime}(\Delta)}{Pred^{k\prime}(\Delta)}\right\}, \tag{3.15}$$

in Step 5. Since (3.14) is a relaxation of the corresponding monotone condition from (3.7), and all other parts in Algorithm 1 remain unchanged (in particular, we still use the condition from (3.6)), it follows that the nonmonotone trust-region method from Algorithm 1 with the update of $\hat{r}^k$ from (3.15) is also well-defined (in particular, the inner loop at each outer iteration k is always finite) and also inherits the local convergence properties from the monotone method and, therefore, the fast convergence rate under the error bound condition from Assumption 1. Hence, it remains to show that our nonmonotone modification does not destroy the global convergence. This is the aim of the next section.

Concerning the fast convergence rate under the error bound condition of Assumption 1, we observe that it has been proved in [92] for the monotone algorithm by showing that the trial direction $\bar{d}^k(\Delta^k)$ satisfies condition (3.6) and (3.7) for k sufficiently large. In our nonmonotone version of the algorithm, condition (3.7) is relaxed. As a consequence, the trial direction $\bar{d}^k(\Delta^k)$ is again accepted for k sufficiently large, and the fast local convergence directly follows from the results stated in [92].

3.3 Global Convergence

Here we want to show that every accumulation point of a sequence generated by the nonmonotone trust-region method from Algorithm 1 with the specification of $\hat{r}^k$ from (3.15) is still a stationary point of the corresponding optimization problem (3.2). A central step in this direction is contained in the following result. Recall that, also in this section, we assume implicitly that our method does not terminate after finitely many iterations with an exact stationary point.

Lemma 2. Let $\{x^k\}$ be a sequence generated by Algorithm 1 with $\hat{r}^k$ defined by (3.15). Then the inequality

$$\Psi(x^0) - \Psi(x^{k+1}) \geq -\rho_1 \sum_{l=0}^{k} Pred_l(\Delta_l^*) \tag{3.16}$$

holds for every $k \in \mathbb{N}$.

Proof. Let $k \geq 0$ be fixed. Then we define an index

$$p(k+1) := \begin{cases} k, & \text{if } \dfrac{Ared^k(\Delta^{k^*})}{Pred^k(\Delta^{k^*})} \geq \dfrac{Ared^{k\,\prime}(\Delta^{k^*})}{Pred^{k\,\prime}(\Delta^{k^*})}, \\ r(k), & \text{otherwise,} \end{cases}$$

where $r(k)$ denotes the index defined in (3.11). We call the corresponding iterate $x^{p(k+1)}$ the "predecessor" of x^{k+1}. We then construct a sequence of strictly increasing iteration counters

$$\{k_0, k_1, k_2, k_3, \ldots\}$$

in such a way that $k_i = p(k_{i+1})$ for all $i = 1, 2, \ldots$, i.e. x^{k_i} is the predecessor of $x^{k_{i+1}}$. Recursively, this means that $k_{i-1} = p(k_i) = p(p(k_{i+1}))$ and so on, until we get down to the index k_1 such that the starting point x^0 is the predecessor of x^{k_1}, i.e. $k_0 := 0 := p(k_1)$. By definition, for each $i \in \mathbb{N}$, we either have $k_i = p(k_{i+1}) = k_{i+1} - 1$ or $k_i = p(k_{i+1}) = r(k_{i+1} - 1)$.

Now, consider the iterate x^{k+1}, and let x^{k_l} be the corresponding predecessor for some $l \in \mathbb{N}$. Then, we can write

$$\begin{aligned} \psi(x^0) - \psi(x^{k+1}) &= \psi(x^0) - \psi(x^{k_1}) && (3.17) \\ &\quad + \sum_{i=1}^{l-1} \left[\psi(x^{k_i}) - \psi(x^{k_{i+1}})\right] && (3.18) \\ &\quad + \psi(x^{k_l}) - \psi(x^{k+1}). && (3.19) \end{aligned}$$

We next take a closer look at each of the three terms (3.17)–(3.19).

First consider the term (3.18) and choose an arbitrary (but fixed) index $i \in \{1, \ldots, l-1\}$. We distinguish two situations: if $k_i = p(k_{i+1}) = k_{i+1} - 1$, we have $Ared_{k_i}(\Delta^*_{k_i})/Pred_{k_i}(\Delta^*_{k_i}) \geq \rho_1$. Since the denominator is negative in view of (3.9), we therefore get

$$\begin{aligned} \psi(x^{k_i}) - \psi(x^{k_{i+1}}) &= \psi(x^{k_i}) - \psi(x^{k_i+1}) \\ &= -Ared_{k_i}(\Delta^*_{k_i}) \\ &\geq -\rho_1 Pred_{k_i}(\Delta^*_{k_i}) \\ &= -\rho_1 \sum_{j=k_i}^{k_{i+1}-1} Pred_j(\Delta^*_j). \end{aligned}$$

On the other hand, if $k_i = p(k_{i+1}) = r(k_{i+1} - 1)$, we have

$$Ared'_{k_{i+1}-1}(\Delta^*_{k_{i+1}-1}) \leq \rho_1 Pred'_{k_{i+1}-1}(\Delta^*_{k_{i+1}-1})$$

because of (3.13) and (3.9). We then obtain

$$\begin{array}{rcl}\psi(x^{k_i}) - \psi(x^{k_{i+1}}) & = & \psi(x^{r(k_{i+1}-1)}) - \psi(x^{k_{i+1}}) \\ & = & \psi_{max}^{k_{i+1}-1} - \psi(x^{k_{i+1}}) \\ & = & -Ared'_{k_{i+1}-1}(\Delta^*_{k_{i+1}-1}) \\ & \geq & -\rho_1 Pred'_{k_{i+1}-1}(\Delta^*_{k_{i+1}-1}) \\ & = & -\rho_1 \displaystyle\sum_{j=r(k_{i+1}-1)}^{k_{i+1}-1} Pred_j(\Delta^*_j) \\ & = & -\rho_1 \displaystyle\sum_{j=p(k_{i+1})}^{k_{i+1}-1} Pred_j(\Delta^*_j) \\ & = & -\rho_1 \displaystyle\sum_{j=k_i}^{k_{i+1}-1} Pred_j(\Delta^*_j). \end{array}$$

In a similar way, we can deal with the two terms from (3.17) and (3.19), respectively, and obtain that

$$\psi(x^0) - \psi(x^{k_1}) \geq -\rho_1 \sum_{j=0}^{k_1-1} Pred_j(\Delta^*_j)$$

and

$$\psi(x^{k_l}) - \psi(x^{k+1}) \geq -\rho_1 \sum_{j=k_l}^{k} Pred_j(\Delta^*_j).$$

The assertion follows by inserting the last three estimates into (3.17)–(3.19). □

The previous result allows us to prove global convergence of the nonmonotone trust-region method.

Theorem 1. Let $\{x^k\}$ be the sequence generated by Algorithm 1 with $\hat{r}^k$ defined by (3.15). Then every accumulation point of $\{x^k\}$ is a stationary point of (3.2).

Proof. Let x^* be an accumulation point of $\{x^k\}$ and let

$$\lim_{k\in K,\, k\to\infty} x^k = x^*$$

be a convergent subsequence. By contradiction, suppose that x^* is not a stationary point of problem (3.2). Then

$$\gamma^* := \lim_{k \in K,\, k \to \infty} \gamma^k = \min\left\{1, \frac{\Delta_{max}}{\|\nabla\Psi(x^*)\|}, \frac{\eta\Psi(x^*)}{\|\nabla\Psi(x^*)\|^2}\right\}$$

is a positive number. The definition of $\bar{d}_G^k(\Delta)$ via (3.4) and (3.5) together with the continuity of the projection operator then implies

$$\left\|\bar{d}_G^k(\Delta_{max})\right\| \to \left\|P_\Omega[x^* - \gamma^*\nabla\Psi(x^*)] - x^*\right\| > 0 \quad \text{as } k \in K, k \to \infty,$$

where the right-hand side is positive since $\gamma^* > 0$ and x^* is not a stationary point. Hence, there exist an index $\tilde{k} > 0$ and a constant $b > 0$ such that

$$\left\|\bar{d}_G^k(\Delta_{max})\right\| \geq b \quad \forall k \in K, k \geq \tilde{k}. \tag{3.20}$$

Moreover, using Proposition 1, there exist $\hat{k}$ and $\bar{\Delta}$ such that, for all $k \in K, k \geq \hat{k}$, and all $\Delta \in (0, \bar{\Delta})$, the two conditions (3.6) and (3.14) are satisfied. Let us define $\hat{k} := \max\{\hat{k}, \tilde{k}\}$.

Let $k \in K$ and $k \geq \hat{k}$. Using the previous observation together with the updating rule of the trust-region radius in Algorithm 1, it follows that

$$\Delta^{k^*} > \alpha_1\bar{\Delta}, \tag{3.21}$$

so we have a uniform lower bound on the size of Δ^{k^*} for all sufficiently large $k \in K$. Taking into account Lemma 2, inequalities (3.9) and (3.6) as well as Lemma 1, we obtain for all $k \in K$, $k \geq \hat{k}$ that

$$\begin{aligned}
\Psi(x^0) - \Psi(x^{k+1}) &\geq -\rho_1 \sum_{l=0}^{k} Pred_l(\Delta_l^*) \\
&\geq -\rho_1 \sum_{l \in K,\, l \geq \hat{k}}^{k} Pred_l(\Delta_l^*) \\
&\geq -\rho_1\sigma \sum_{l \in K,\, l \geq \hat{k}}^{k} \nabla\Psi(x_l)^T \bar{d}_l^G(\Delta_l^*) \\
&\geq \rho_1\sigma \sum_{l \in K,\, l \geq \hat{k}}^{k} \Big(\frac{\Delta_l^*}{\Delta_{max}\gamma_l}\Big) \|\bar{d}_l^G(\Delta_{max})\|^2 \\
&\geq \rho_1\sigma \sum_{l \in K,\, l \geq \hat{k}}^{k} \Big(\frac{\Delta_l^*}{\Delta_{max}}\Big) b^2.
\end{aligned} \tag{3.22}$$

where the last inequality follows from (3.20) and the fact that $\gamma^k \leq 1$. Taking $k \to \infty$ with $k \in K$, the right-hand side of (3.22) goes to infinity because of (3.21), while the left-hand side is bounded since $\Psi(x) \geq 0$ for all $x \in \mathbb{R}^n$. This contradiction completes the proof. □

3.4 Application to Generalized Nash Equilibrium Problems

As noted in the introduction, a typical class of problems which often have nonunique solutions are the so-called Generalized Nash Equilibrium Problems (GNEPs). Let N be the number of players (or agents) ν of the GNEP ($\nu = 1, ..., N$). Each agent's problem then consists of an optimization problem of the form

$$\min_{z_\nu} \;\; \theta_\nu(z_\nu, z_{-\nu}) \quad \text{s.t.} \quad g_\nu(z_\nu, z_{-\nu}) \leq 0, \tag{3.23}$$

where $z_\nu \in \mathbb{R}^{n_\nu}$ represents the variables which are controlled by the ν-th player, $z_{-\nu} \in \mathbb{R}^{n-n_\nu}$ express the remaining ones he cannot control, $\theta_\nu : \mathbb{R}^n \to \mathbb{R}$ is his objective (or utility) function, while $g_\nu : \mathbb{R}^n \to \mathbb{R}^{m_\nu}$ describes his constraints and defines the possible strategies of player ν. Note that both θ_ν and g_ν depend on all variables, but that (3.23) is an optimization problem in z_ν only (parameterized by $z_{-\nu}$).

The ν-th player controls n_ν variables and the total number of variables of the problem, grouped under the name z, is $n := n_1 + \cdots + n_N$. Without loss of generality, we assume that players' options are only constrained by inequalities. The decision of player ν is affected by m_ν inequalities, and the total number of inequalities of the problem is $m := m_1 + \cdots + m_N$.

A solution of a GNEP is called *generalized Nash equilibrium* and is reached when none of the N agents is able to obtain a better value for his utility function by unilaterally changing his strategy. This means that $z^* \in \mathbb{R}^n$ is a generalized Nash equilibrium if, for all $\nu = 1, ..., N$, it holds that

$$g_\nu(z^*_\nu, z^*_{-\nu}) \leq 0 \text{ and } \theta_\nu(z^*_\nu, z^*_{-\nu}) \leq \theta_\nu(z_\nu, z^*_{-\nu}) \qquad \forall z_\nu \in \mathbb{R}^{n_\nu} : \; g_\nu(z_\nu, z^*_{-\nu}) \leq 0.$$

We assume the following properties to hold for all θ_ν and g_ν.

Assumption 2. Functions θ_ν and g_ν are twice continuously differentiable with locally Lipschitz continuous second order derivatives for all $\nu = 1, ..., N$.

Assumption 3. The GNEP is player-convex, i.e. $\theta_\nu(\cdot, z_{-\nu})$ and $g_{\nu,i}(\cdot, z_{-\nu})$ are convex functions for every $\nu = 1, ..., N$, $i = 1, ..., m_\nu$ and $z_{-\nu}$.

Note that the player-convex case is typically the most general class of GNEPs considered in the literature and, in particular, allows much more freedom than the jointly-convex case, cf. [35] for more details.

The Lagrangian function related to the ν-th optimization problem (3.23) is

$$L_\nu(z, \lambda_\nu) := \theta_\nu(z_\nu, z_{-\nu}) + \sum_{i=1}^{m_\nu} \lambda_{\nu,i} g_{\nu,i}(z_\nu, z_{-\nu}),$$

where $\lambda_\nu \in \mathbb{R}^{m_\nu}$ is the vector of Lagrange multipliers of player ν. If we concatenate all the multipliers, all the constraints, and all the gradients of the Lagrangian, we obtain

$$\lambda := \big(\lambda_\nu\big)_{\nu=1}^N, \qquad H(z, \lambda) := \big(\nabla_{z_\nu} L_\nu(z, \lambda_\nu)\big)_{\nu=1}^N, \qquad g(z) := \big(g_\nu(z)\big)_{\nu=1}^N.$$

Let z be a solution of (3.23). If we assume any standard constraint qualification to hold for every player $\nu = 1, \ldots, N$, it follows that there exists a vector $\lambda \in \mathbb{R}^m$ such that the following (concatenated) KKT-system holds:

$$H(z, \lambda) = 0, \qquad \lambda \geq 0, \qquad g(z) \leq 0, \qquad \lambda^T g(z) = 0. \tag{3.24}$$

Conversely, any solution of the system yields a solution of the GNEP (without any constraint qualification) due to the assumed player-convexity property.

We, therefore, focus on solving (3.24) in order to obtain a solution of the original problem (3.23). The mixed system (3.24), in turn, might be reformulated in different ways, but for reasons explained in the introduction, we concentrate on a simple, smooth reformulation as a constrained nonlinear system of equations: introducing slack variables $w \in \mathbb{R}^m_+$ and using the Hadamard product (the component-wise product $(w \circ \lambda)_i := w_i \lambda_i$ for all $i = 1, \ldots, m$), we obtain the following box-constrained system of equations:

$$F(x) := \begin{pmatrix} H(z, \lambda) \\ g(z) + w \\ w \circ \lambda \end{pmatrix} = 0, \qquad s.t. \quad x := (z, \lambda, w) \in \Omega := \mathbb{R}^n \times \mathbb{R}^m_+ \times \mathbb{R}^m_+. \tag{3.25}$$

We assume that the corresponding solution set $X^* := \{x \in \Omega : F(x) = 0\}$ is nonempty.

Then we are precisely in the situation described in (3.1). Since Ω is defined by some nonnegativity constraints only, the projections onto Ω are easy to compute. Assumption 1 (a)-(c) also hold in our GNEP-setting. Furthermore, also the local error bound condition from (d) of this assumption was shown to be satisfied under suitable assumptions in [31]. In particular, these assumptions do not require that the Jacobian of F is nonsingular at a solution. In fact, even nonisolated solutions are allowed. We do not recall the precise conditions here and refer the interested reader to [31] for more details.

3.5 Numerical Results

In this section we report some numerical results of computational experiments performed in order to evaluate the effectiveness of the proposed nonmonotone algorithm. To this end, we first recall that our method is particularly designed to solve difficult problems with high accuracy under a relatively weak (error bound) condition. For those examples which satisfy stronger assumptions like a nonsingularity condition, one has to expect that other Newton-type schemes are more efficient. Nevertheless, the overall results indicate that our method is very competitive.

Our first aim is to compare the numerical behavior of our method on a larger set of examples. To this end, we use the following three algorithms:

1. Algorithm 1 with r^k defined in (3.8): the original Monotone Trust-Region (MTR) framework by Tong and Qi [92];
2. Algorithm 1 with r^k defined in (3.15): the Nonmonotone Trust-Region (NTR) framework extending MTR;
3. STRSCNE [7, 8], a specific solver for nonlinear, box-constrained systems of equations.

MTR and NTR have been implemented in MATLAB, the same programming language used also for the STRSCNE code. The algorithms terminate successfully when

$$\|F(x^k)\|_\infty \leq \epsilon_1, \qquad \text{with } \epsilon_1 = 10^{-4}. \tag{3.26}$$

For the algorithms, the maximum numbers of function evaluations and Jacobian evaluations have been fixed equal to 100,000. The termination criterion described in Algorithm 1 ($\|\bar{d}_G^k(\Delta)\| < \epsilon_2$) and used in our theoretical investigation has not been employed in our implementation since STRSCNE does not use any similar criterion. Regarding nonmonotone window, above results are obtained with $W = 50$, thus NTR might be considered strongly nonmonotone.

The three methods have been tested on a dataset of 35 different GNEPs [32]. The total number of runs is 57 because multiple starting points have been used for some of the problems (see Table 5.1 of [32] for more details).

We first compare the number of failures obtained by the three algorithms:

- 9 for STRSCNE;
- 12 for MTR;
- 7 for NTR.

From the above comparison, we might see that the adoption of the nonmonotone strategy yields significant advantages in terms of computed global solutions, with respect to the original monotone version of Algorithm 1 (MTR). In fact, MTR seems to converge more frequently to nonoptimal stationary points. Furthermore, Algorithm NTR also outperforms STRSCNE.

The algorithms have been numerically compared using performance profiles [30]. In Figure 3.1 we show results obtained employing the following measures:

- total number of function evaluations (Figure 3.1(a));
- total number of Jacobian evaluations (Figure 3.1(b));
- actual computational time, estimated by `tic toc` MATLAB function (Figure 3.1(c)).

Note that in order to obtain reliable measurements on computational time, experiments have been repeated several times, choosing the minimum obtained results as the final one. In addition, algorithms are considered to be equivalent on this metric, if the difference between their measures is smaller than $10^{-3}\, s$. Moreover, note that scale used for all the metrics in Figure 3.1 is logarithmic since results are sometimes pretty close.

Figure 3.1 clearly shows the good performances of the proposed algorithm with respect to all the employed metrics. Again, it may be observed that the employment of the nonmonotone strategy leads to a significant improvement

Table 3.1: Detailed numerical results for problem A11.

k	$\|F(x)\|_\infty$	$\|\bar{d}_G^k(\Delta^{k^*})\|_2$	Δ^k
0	1.00e+02	1.1e-01	5.00
1	5.65e+01	2.0e-01	10.00
2	1.56e+01	2.6e-01	20.00
3	3.85e+00	1.5e-01	40.00
4	9.48e-01	1.5e-01	80.00
5	2.22e-01	4.2e-02	160.00
6	4.33e-02	1.1e-02	320.00
7	4.44e-03	2.0e-03	640.00
8	7.38e-05	5.2e-05	1000.00
Final	2.20e-08	3.1e-08	

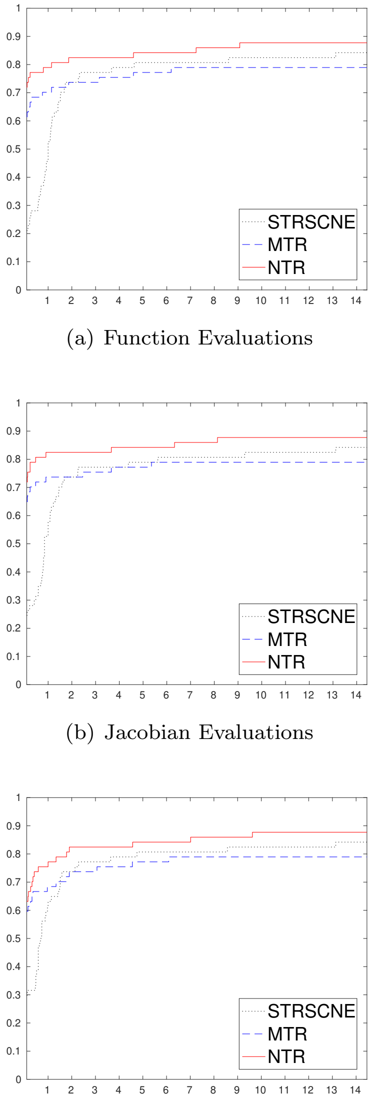
of the performances of the original monotone version of the algorithm. The proposed algorithm may be considered at least competitive with a sound and efficient code as STRSCNE.

An illustrative example

Here, we illustrate the performance obtained by NTR on a specific GNEP from [32]: example A11. As presented in *Example 1* from [31], this is a peculiar problem because the error bound condition for (3.25) is proven to be valid for the whole solution set, while the Jacobian of F is singular at all solutions. This means that local fast convergence for classical Newton-like methods cannot be expected here, while MTR and NTR are proven to be locally superlinearly convergent.

In Table 3.1 we present the iteration history obtained by NTR where the columns show, for each iteration, $\{\|F(x^k)\|_\infty\}$, $\{\|\bar{d}_G^k(\Delta^{k^*})\|_2\}$ and the initial trust-region radius Δ^k.

From Table 3.1 we may observe the two phases of the algorithm: the globalization strategy might be considered active between iterations 0 and 4, as the measure of stationarity $\|\bar{d}_G^k(\Delta^{k^*})\|$ is not sufficiently small and not always decreasing. Starting with iteration 5, the norm of the projected gradient is decreasing rapidly, following the typical behavior of superlinear convergence.



(a) Function Evaluations

(b) Jacobian Evaluations

(c) Time

Figure 3.1: Performance profiles between STRSCNE (black dotted line), MTR (blue dashed line) and NTR (red solid line) on GNEP library.

3.5.1 Experiment on Quasi-Variational Inequalities

In this subsection, we present some additional numerical experiments on a different class of problems, QVIs. QVIs are similar to GNEPs in many sense. First of all, a GNEP might be reformulated as a QVI under some convexity assumptions. The feasible set of both problems is not known in advance: it changes with respect to values of some variables of the problem. From a practical point of view, QVI might be reformulated using KKT in a way that is similar to that expressed in Section 3.4. As described in detail in Section 2 from [37], this procedure leads to a nonlinear system of equations that has the same structure of (3.25): this is the system on which the three algorithms NTR, MTR and STRSCNE are going to be tested in this subsection.

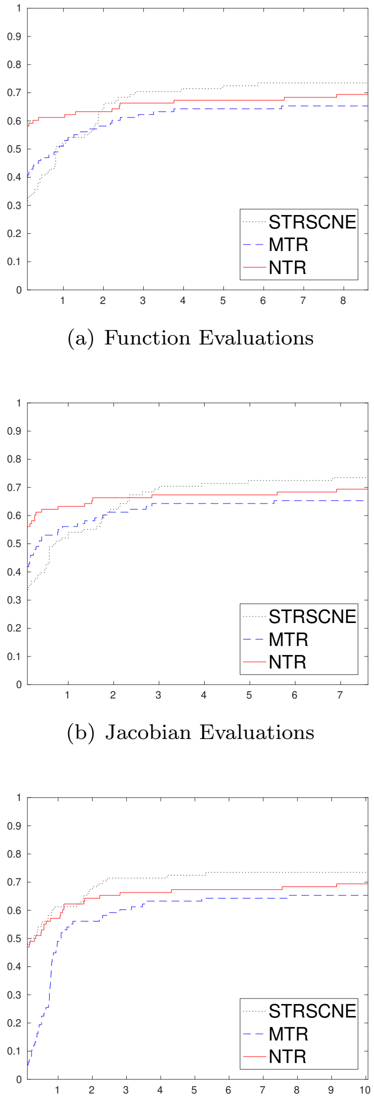
On the other hand, from the theoretical point of view, there is also an important difference between these two classes of problems: no local error bound results have been proven for QVIs. For the class of QVIs, superlinear convergence is only proven under some nonsingularity assumptions. Thus, when this assumption is satisfied, all other Newton-type schemes are able to obtain fast local convergence.

In this subsection, we consider a set of QVIs which is taken from a library called QVILIB [37]. QVILIB is composed by 55 different problems. For each of them, two (in few cases more than two) different starting points are available, leading to a total of 116 different executions. In this work, we focus on the subset of QVIs for which the associated KKT system is smaller than 10000 equations: reaching high accuracy on big problems might be considered a totally different topic and it goes beyond the scope of this work. The total number of executions in the new set is, thus, reduced to 98.

The setting used to face QVIs is exactly the same as the one used for GNEPs, except for 2 elements: an additional termination criterion that checks the maximum computational time (1 hour) and a difference in the dimension of the nonmonotone window. In fact in this case, best results are obtained with $W = 10$. Thus, NTR might be considered lightly nonmonotone. Algorithms terminate successfully exactly as described in (3.26).

In Figure 3.2, we show performance profiles obtained employing the same measures as in Figure 3.1. The total number of failures obtained by the three algorithms is:

- 26 for STRSCNE;



(a) Function Evaluations

(b) Jacobian Evaluations

(c) Time

Figure 3.2: Performance profiles between STRSCNE (black dotted line), MTR (blue dashed line) and NTR (red solid line) on QVI library.

- 33 for MTR;
- 29 for NTR.

Figure 3.2, together with the above summary, clearly shows that NTR outperforms MTR with respect to all measures we considered and with respect to the number of failures. Thus, even with a small window, the nonmonotone extension proposed has been showed to be crucial in order to avoid nonoptimal critical points and improve performances.

Regarding comparison with STRSCNE, we might note that NTR performs better on the majority of the problems, both with respect to function and Jacobian evaluations, till $\tau \leq 4$. On time evaluation the two methods have similar performances, again till $\tau \leq 4$. When we take into account greater τ, considering stability of the method, STRSCNE seems to be more reliable. This is actually supported also by the comparison on the total number of failures.

On the other hand, as already observed in the latter, in the field of QVI there aren't results that prove a local error bound condition, while Tong and Qi's method have been mainly taken into account in order to achieve fast convergence under such a condition only. On the contrary, STRSCNE is a stable software that often exploits direct Newton steps and does not cope with error bound conditions.

In conclusion, regarding the overall number of failures, recall that QVIs is a class of problems which is known to be very difficult to solve, especially by Newton-type methods and similar, as the three algorithms above.

3.6 Conclusions

This work modifies a (monotone) trust-region method for constrained systems of equations by introducing a suitable nonmonotonicity criterion. The corresponding nonmonotone trust-region method is shown to be globally convergent to stationary points and locally fast convergent under an error bound condition. Numerical results obtained for a suitable reformulation of generalized Nash equilibrium problems indicate that the nonmonotone method is both more reliable and more efficient than its monotone counterpart. Furthermore, some additional tests performed on quasi-variational inequalities confirm aforementioned improvements of the nonmonotone extension. A preliminary testing shows that the numerical behavior can still be improved if one simply skips the additional criterion (3.6) from Algorithm 1. Part of our future research is, therefore, to investigate whether this condition can (at least) be relaxed in an appropriate way. Regarding QVIs, the lack of a local error bound result might actually be something that deserves some attention in our future researches.

3.7 Proofs of Lemma 1 and Proposition 1

We first recall some elementary properties of the projection operator.

Lemma 3. The following statements hold:

(a) $\big(P_\Omega(z) - z\big)^T \big(P_\Omega(z) - x\big) \leq 0 \quad \forall x \in \Omega,\ \forall z \in \mathbb{R}^n$;

(b) $\|P_\Omega(x_2) - P_\Omega(x_1)\| \leq \|x_2 - x_1\| \quad \forall x_1, x_2 \in \mathbb{R}^n$.

(c) Given $x, d \in \mathbb{R}^n$, the function

$$\theta(t) := \frac{\big\|P_\Omega(x + td) - x\big\|}{t}, \quad t > 0,$$

is nonincreasing.

The first two properties in Lemma 3 are a well-known characterization of the projection, whereas the third property was shown, e.g., in [10] in the context of a suitable globalization of a projected gradient method.

Proof. of Lemma 1: Let $k \in \mathbb{N}$ be fixed, choose $\Delta > 0$, and define

$$z^k := x^k + d_G^k(\Delta) = x^k - \frac{\Delta}{\Delta_{\max}} \gamma^k \nabla\Psi(x^k)$$

for the sake of notational convenience. Then an elementary calculation yields

$$\begin{aligned}
\nabla\Psi(x^k)^T \bar{d}_G^k(\Delta) &= \nabla\Psi(x^k)^T \big(P_\Omega(z^k) - x^k\big) \\
&= \frac{\Delta_{\max}}{\Delta \cdot \gamma^k} \big(x^k - z^k\big)^T \big(P_\Omega(z^k) - x^k\big) \\
&= \frac{\Delta_{\max}}{\Delta \cdot \gamma^k} \big(P_\Omega(z^k) - z^k\big)^T \big(P_\Omega(z^k) - x^k\big) \\
&\quad + \frac{\Delta_{\max}}{\Delta \cdot \gamma^k} \big(x^k - P_\Omega(z^k)\big)^T \big(P_\Omega(z^k) - x^k\big) \\
&\leq -\frac{\Delta_{\max}}{\Delta \cdot \gamma^k} \big\|\bar{d}_G^k(\Delta)\big\|^2,
\end{aligned}$$

where the inequality follows from Lemma 3 (a), the definition of $\bar{d}_G^k(\Delta)$, and the feasibility of x^k. On the other hand, Lemma 3 (c) with $d := -\nabla\Psi(x^k)$ implies that

$$\begin{aligned}
\frac{\|\bar{d}_G^k(\Delta)\|}{\Delta} &= \frac{\|P_\Omega\big(x^k - \frac{\Delta}{\Delta_{\max}}\gamma^k\nabla\Psi(x^k)\big) - x^k\|}{\Delta} \\
&\geq \frac{\|P_\Omega\big(x^k - \gamma^k\nabla\Psi(x^k)\big) - x^k\|}{\Delta_{\max}} \\
&= \frac{\|\bar{d}_G^k(\Delta_{\max})\|}{\Delta_{\max}}
\end{aligned}$$

holds for all $0 < \Delta \leq \Delta_{\max}$. Combining the last two inequalities yields the assertion. □

Proof. of Proposition 1: Since $x^k \to x^*$ for $k \in K$ and $k \to \infty$, the continuity of F' implies that there is a constant b_1 such that $\|F'(x^k)\| \leq b_1$ for all $k \in K$. Using (3.4), (3.5), and Lemma 3 (b), we therefore obtain for all $k \in K$

$$\begin{aligned}
\|F'(x^k)\bar{d}_G^k(\Delta)\| &= \|F'(x^k)(P_\Omega[x^k + d_G^k(\Delta)] - x^k)\| \\
&\leq \|F'(x^k)\|\|x^k + d_G^k(\Delta) - x^k\| \\
&\leq \frac{\Delta\gamma^k}{\Delta_{max}}\|F'(x^k)\|\|\nabla\Psi(x^k)\| \\
&\leq b_1\Delta,
\end{aligned} \tag{3.27}$$

where the last inequality follows from the definition of γ^k in Step 2.

Since x^* is not a stationary point by assumption, we can follow the argument from the first part of the proof of Theorem 1 in order to see that there is a constant $b > 0$ such that

$$\|\bar{d}_G^k(\Delta_{max})\| \geq b \quad \forall k \in K, k \geq \hat{k}. \tag{3.28}$$

Let

$$\tilde{\Delta} = \min\left\{\Delta_{max}, \frac{(1-\sigma)b^2}{b_1^2\Delta_{max}}\right\}. \tag{3.29}$$

We first prove that (3.6) holds for all $k \in K, k \geq \hat{k}$ and all $\Delta \in (0, \tilde{\Delta}]$. From the definition of $\bar{d}^k(\Delta)$, we get that

$$\begin{aligned}
Pred^k(\Delta) &= \frac{1}{2}\big\|F(x^k) + F'(x^k)\bar{d}^k(\Delta)\big\|^2 - \Psi(x^k) \\
&\le \frac{1}{2}\big\|F(x^k) + F'(x^k)\bar{d}^k_G(\Delta)\big\|^2 - \Psi(x^k) \\
&= \nabla\Psi(x^k)^T\bar{d}^k_G(\Delta) + \frac{1}{2}\big\|F'(x^k)\bar{d}^k_G(\Delta)\big\|^2 \\
&= \sigma\nabla\Psi(x^k)^T\bar{d}^k_G(\Delta) + (1-\sigma)\nabla\Psi(x^k)^T\bar{d}^k_G(\Delta) \\
&\quad + \frac{1}{2}\big\|F'(x)\bar{d}^k_G(\Delta)\big\|^2 \\
&\le \sigma\nabla\Psi(x^k)^T\bar{d}^k_G(\Delta) - (1-\sigma)\Big(\frac{\Delta}{\Delta_{max}\gamma^k}\Big)\big\|\bar{d}^k_G(\Delta_{max})\big\|^2 + \frac{1}{2}b_1^2\Delta^2 \\
&\le \sigma\nabla\Psi(x^k)^T\bar{d}^k_G(\Delta) - b_1^2\Delta\tilde{\Delta} + \frac{1}{2}b_1^2\Delta^2 \\
&\le \sigma\nabla\Psi(x^k)^T\bar{d}^k_G(\Delta),
\end{aligned}$$

where the second inequality follows directly from Lemma 1 and (3.27), the third inequality follows from (3.28) and (3.29) and recalling that $0 < \gamma^k \le 1$, while the last inequality holds since $\Delta \le \tilde{\Delta}$.

In order to prove that (3.7) holds for $k \in K$ and k sufficiently large and for Δ belonging to an interval $(0, \bar{\Delta}]$, we will first show that

$$-Pred^k(\Delta) \ge \beta\Delta, \tag{3.30}$$

and

$$Ared^k(\Delta) - Pred^k(\Delta) \le c_1\Delta^2 \tag{3.31}$$

hold for suitable constants $\beta > 0$ and $c_1 > 0$

First we show that (3.30) holds. To this aim, taking $\Delta \in (0, \tilde{\Delta}]$, using

Lemma 1 and (3.27), we can write

$$\begin{aligned}
\frac{1}{2}\left\|F(x^k)+F'(x^k)\bar{d}_G^k(\Delta)\right\|^2 &= \frac{1}{2}\|F(x^k)\|^2 + \nabla\Psi(x^k)^T\bar{d}_G^k(\Delta) \\
&\quad + \frac{1}{2}\left\|F'(x^k)\bar{d}_G^k(\Delta)\right\|^2 \\
&\leq \Psi(x^k) - \Big(\frac{\Delta}{\gamma^k\Delta_{max}}\Big)\left\|\bar{d}_G^k(\Delta_{max})\right\|^2 \\
&\quad + \frac{1}{2}b_1^2\Delta^2 \\
&\leq \Psi(x^k) - \Big(\frac{\Delta}{\gamma^k\Delta_{max}}\Big)\left\|\bar{d}_G^k(\Delta_{max})\right\|^2 \\
&\quad + \frac{1}{2}\Delta\frac{b^2(1-\sigma)}{\Delta_{max}} \\
&\leq \Psi(x^k) - \Big(\frac{\Delta}{\gamma^k\Delta_{max}}\Big)\left\|\bar{d}_G^k(\Delta_{max})\right\|^2 \\
&\quad + \frac{1}{2}\Delta\frac{\left\|\bar{d}_G^k(\Delta_{max})\right\|^2}{\gamma^k\Delta_{max}} \\
&= \Psi(x^k) - \Big(\frac{\Delta}{2\gamma^k\Delta_{max}}\Big)\left\|\bar{d}_G^k(\Delta_{max})\right\|^2,
\end{aligned}$$

where the second inequality follows from (3.29), and the third holds recalling that $(1-\sigma)<1, \gamma^k \leq 1$, and (3.28). Consequently, we have

$$\begin{aligned}
Pred^k(\Delta) &\leq \frac{1}{2}\left\|F(x^k)+F'(x^k)\bar{d}_G^k(\Delta)\right\|^2 - \Psi(x^k) \\
&\leq -\left(\frac{\Delta}{2\gamma^k\Delta_{max}}\right)\left\|\bar{d}_G^k(\Delta_{max})\right\|^2 < 0,
\end{aligned}$$

where the first inequality follows from the definitions of $\bar{d}^k$ and $t^*(\Delta)$ in Step 4 Thus, using (3.28) and recalling again that $\gamma^k \leq 1$, we obtain that there exists $\beta > 0$ such that (3.30) is satisfied for all $k \in K, k \geq \hat{k}$ and all $\Delta \in (0, \tilde{\Delta}]$.

Now we prove (3.31). From Lemma 3 (b), recalling the definitions of $d_G^k(\Delta)$, $\bar{d}_G^k(\Delta)$ and γ^k, we have

$$\left\|\bar{d}_G^k(\Delta)\right\| \leq \left\|d_G^k(\Delta)\right\| \leq \Delta, \quad \forall\Delta \in (0, \Delta_{max}].$$

From the definition of $\bar{d}_{tr}^k(\Delta)$, using Lemma 3 (b) again, and recalling that $d_{tr}^k(\Delta)$ is the trust-region solution, we have

$$\left\|\bar{d}_{tr}^k(\Delta)\right\| \leq \left\|d_{tr}^k(\Delta)\right\| \leq \Delta, \quad \forall\Delta \in (0, \Delta_{max}].$$

Consequently, from the last two inequalities we get $\|\bar{d}^k(\Delta)\| \le \Delta$. Since F' is locally Lipschitzian, it is globally Lipschitz on compact sets. Consequently, $\nabla\Psi$ is also globally Lipschitz on compact sets. Since $x^k \to x^*$ for $k \in K$ and $\bar{d}^k(\Delta)$ is bounded for all $\Delta \in (0, \Delta_{\max}]$, we can apply the Mean Value Theorem and obtain the existence of suitable numbers $\theta^k \in (0,1)$ and a Lipschitz constant $L > 0$ such that

$$\begin{aligned}\Psi\big(x^k + \bar{d}^k(\Delta)\big) - \Psi(x^k) - \nabla\Psi(x^k)^T\bar{d}^k(\Delta) &= \nabla\Psi\big(x^k + \theta^k\bar{d}^k(\Delta)\big)^T\bar{d}^k(\Delta) \\ &\quad - \nabla\Psi(x^k)^T\bar{d}^k(\Delta) \\ &\le L\Delta\|\bar{d}^k(\Delta)\|\end{aligned}$$

for all $k \in K$ and all $\Delta \in (0, \Delta_{\max}]$, where the last inequality takes into account the Cauchy-Schwarz inequality. Hence, we can write

$$\begin{aligned}Ared^k(\Delta) - Pred^k(\Delta) &= \Psi\big(x^k + \bar{d}^k(\Delta)\big) - \frac{1}{2}\big\|F(x^k) + F'(x^k)\bar{d}^k(\Delta)\big\|^2 \\ &= \Psi\big(x^k + \bar{d}^k(\Delta)\big) - \Psi(x^k) - \nabla\Psi(x^k)^T\bar{d}^k(\Delta) \\ &\quad - \frac{1}{2}\bar{d}^k(\Delta)^T F'(x^k)^T F'(x^k)\bar{d}^k(\Delta) \\ &\le L\Delta\big\|\bar{d}^k(\Delta)\big\| - \frac{1}{2}\bar{d}^k(\Delta)^T F'(x^k)^T F'(x^k)\bar{d}^k(\Delta) \\ &\le L\Delta^2 + c_2\big\|\bar{d}^k(\Delta)\big\|^2 \\ &\le c_1\Delta^2\end{aligned}$$

for suitable constants $c_1, c_2 > 0$.

Finally, exploiting (3.30) and (3.31), it follows that there exists $\bar{\Delta} > 0$ such that

$$\hat{r}^k = 1 - \frac{Ared^k(\Delta) - Pred^k(\Delta)}{-Pred^k(\Delta)} \ge \rho_1, \quad \forall k \in K, k \ge \hat{k} \text{ and } \forall\Delta \in (0, \bar{\Delta}].$$

This concludes the proof.

□

Chapter 4

A Unified Convergence Framework for Nonmonotone Inexact Decomposition Methods

In this chapter we propose a general framework that provides a unified convergence analysis for nonmonotone decomposition algorithms. The main motivation to embed nonmonotone strategies within a decomposition approach lies in the fact that enforcing the reduction of the objective function could be unnecessarily expensive, taking into account that groups of variables are individually updated. We define different search directions and line searches satisfying the conditions required by the presented nonmonotone decomposition framework to obtain global convergence. We employ a set of large-scale network equilibrium problems as a computational example to show the advantages of a nonmonotone algorithm over its monotone counterpart. In conclusion, a new smart implementation for decomposition methods has been derived to solve numerical issues on large-scale partially separable functions.[1]

[1] This chapter has been published as "A Unified Convergence Framework for Nonmonotone Inexact Decomposition Methods" in *Computational Optimization and Application vol. 75, iss. 1, pp. 113-144. 2019* [45].

4.1 Introduction

Let us consider the problem

$$\begin{aligned} &\min_x f(x), \\ &\text{s.t. } x \in \mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2 \times \cdots \times \mathcal{F}_L \subset \mathbb{R}^n, \end{aligned} \tag{4.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, $\mathcal{F}_h \subseteq \mathbb{R}^{n_h}$, with $h \in \{1, \ldots, L\}$, are compact, nonempty and convex sets, with $n_1 + \cdots + n_h + \cdots + n_L = n$. Note that $x_{(h)} \in \mathbb{R}^{n_h}$ denotes the h-th block component of x, i.e.,

$$x = \left(x_{(1)}, x_{(2)}, \ldots, x_{(h)}, \ldots, x_{(L)}\right)^T .$$

We say that a point $x^\star \in \mathcal{F}$ is a critical point for problem (4.1) if

$$\nabla f(x^\star)^T (x - x^\star) \geq 0 \qquad \forall x \in \mathcal{F}.$$

We assume that the dimension n_h of each block h is very large, so that we are interested in studying decomposition methods, whose strategy is that of iteratively solving a sequence of smaller and simpler subproblems.

Convergence theory for decomposition methods is consolidated when the feasible set is the Cartesian product of closed, nonempty convex subsets (see [9]). More recently, decomposition methods have been studied and designed for some specific classes of optimization problems like Network Equilibrium (NE) problems [29] and training of Support Vector Machines (SVM) [60,72].

The most common decomposition algorithm is the Gauss-Seidel method, where each h-th block component of each iterate x^k is sequentially considered: at each iteration, $x^k_{(h)}$ is updated by solving the corresponding subproblem

$$x^{k+1}_{(h)} \in \arg \min_{\xi_{(h)} \in \mathcal{F}_h} f(x^{k+1}_{(1)}, x^{k+1}_{(2)}, \ldots, \xi_{(h)}, x^k_{(h+1)}, \ldots, x^k_{(L)}).$$

In principle, the method requires to solve exactly each subproblem, but this may be expensive or prohibitive whenever the closed-form solution of the subproblem is not available or the objective function is nonconvex. To overcome this computational issue, block-descent methods have been proposed (see, e.g., [56], [14]). In particular, to update each block component $x^k_{(h)}$ they employ one or more iterations of a line search based algorithm. In contrast to the original Gauss-Seidel method, these methods are called "inexact" because the requirement of computing the global minimum point of each subproblem is relaxed.

In block-descent methods, starting from a feasible initial step length, the backtracking line search procedure reduces the step length until a condition of sufficient reduction of the objective function is satisfied. Thus, the objective function is forced to decrease at each iteration. However, the condition of monotone decrease may frequently lead to not accept a “good” initial step length, i.e., a step length that is showed to obtain good performances in practice. In turn, this often reduces the effectiveness and the efficiency of the method, especially when the problem is highly nonlinear. This motivates the adoption of nonmonotone strategies, where the condition of sufficient reduction is relaxed to accept more frequently a good initial step length.

Note that the literature on nonmonotone methods is wide and concerns several classes of optimization problems including unconstrained problems [53,55,82]), convex constrained problems [11], equality-constrained nonsmooth problems [69], nonlinear complementary problems [27], generalized Nash equilibrium problems [46], derivative-free problems [54].

In this work, we focus on the design and the development of nonmonotone decomposition methods. The main motivations to couple nonmonotone and decomposition strategies are the following:

- the block components are individually updated according to a decomposition strategy. In fact in each block update, it could be unfruitful and unnecessarily expensive to enforce a strict reduction of the objective function, especially when the current iterate is far from a stationary point;
- as already said, the nonmonotone strategy may improve the effectiveness and the efficiency of the method used to solve inexactly the generated subproblems.

In the literature, nonmonotone strategies have already been applied together with decomposition methods to face some specific problems, e.g., training of Support Vector Machines [83, 99]. To the best of our knowledge, this is the first work in which decomposition methods and nonmonotone strategies are combined in a globally convergent framework built in the context of general constrained optimization. The proposed nonmonotone framework extends the existing monotone decomposition framework presented in [16]. In that work, different line search based methods can be embedded and possibly combined with a column generation strategy[2]. Col-

[2] The column generation is the technique that lets optimization algorithms keep only a

umn generation strategies are not considered here, since the focus of this work is the definition of a nonmonotone decomposition framework; however, the theory developed here might be easily extended to the use of a column generation technique thanks to few modifications presented in [16].
The contributions of the present work are the following:

(i) a unifying nonmonotone decomposition framework has been defined;

(ii) new convergence results have been stated for decomposition methods applied to a specific class of optimization problems: problems with a single linear equality constraint and box constraints;

(iii) a smart implementation of nonmonotone line search methods has been suitably defined. In particular, the aim is facing numerical issues arising in the optimization of large-scale partially separable ([48,49]) problems.

Regarding point (ii), convergence results are stated for decomposition methods that employ a search direction based on the Gauss-Southwell rule [93]: a search direction with only two nonzero components, the pair of components that mostly violates the optimality conditions. Similar convergent decomposition methods based on inexact line searches have been recently proposed in [29] and [13]. Concerning point (iii), we observe that large-scale problems often come together with a consistent loss of significant digits, due to operations computed in floating-point arithmetic. For example, subtracting two big and nearly similar numbers might lead to a significant loss of precision called catastrophic cancellation. We present some numerical tricks, for the monotone and nonmonotone case, to take into account these issues and to reduce the loss of precision. The nonmonotone trick was never proposed in the literature before, and allowed us to obtain relevant benefits both in terms of computational time and numerical precision.

The chapter is organized as follows: in the next section a simplified version of the monotone framework from [16] is derived to understand from which point the new theory will start off; in Section 4.3 the whole new framework is described and, in particular, Subsection 4.3.1 and 4.3.2 are respectively dedicated to line search mappings and direction choices; in Section 4.4 the derivation of the new numerical technique is fully presented; in Section 4.5 we take into account a set of large-scale NE problems as an illustrative example to show two facts: the numerical advantages obtained by the

subset of variables in memory, while new variables are iteratively added when needed to reach optimality. See [47] for more details.

nonmonotone algorithm over its monotone counterpart and a detailed analysis of the advantages achieved by employing the new numerical technique; conclusions are outlined in Section 4.6.

4.2 The Original Monotone Decomposition Framework

In this section we report a simplified version of the framework from [16]. The Inexact Decomposition Algorithm (IDA) is sketched below, note that the column generation (Step 4 of the original Algorithm 1 from [16]) has been removed. The main idea behind [16] is that of proving global convergence without directly defining each step of Algorithm 2. In fact, Algorithm 2 is fully characterized when Steps 3 - 5 are defined to satisfy Assumptions 4 - 6 below.

Algorithm 2: Inexact Decomposition Algorithm (IDA)

```
  Input: $x^0 \in \mathcal{F}$ starting point
1 $k = 0$
2 while stopping criterion is not fulfilled do
3     choose $h^k \in \{1, \dots, L\}$
4     define a feasible descent direction such that $d^k_{(h)} = 0_{(h)} \quad \forall h \neq h^k$
5     compute a feasible step length $\alpha^k$ by means of a suitable line
        search along the direction $d^k$
6     $x^{k+1} = x^k + \alpha^k d^k$
7     $k = k + 1$
```

Assumption 4. There exists an integer $M > 0$ such that, for all $k \geq 0$ and for all $h \in \{1, \dots, L\}$, we can find an index $l(k)$ with $0 \leq l(k) \leq M$, such that at Step 3 we have $h^{k+l(k)} = h$.

Assumption 5. At every iteration k, the line search procedure computes a value α^k such that

$$f(x^k + \alpha^k d^k) \leq f(x^k).$$

Furthermore, if $\{x^k\}$ is a sequence of feasible points convergent to a point $\bar{x}$

and

$$\lim_{k\to\infty} f(x^k) - f(x^k + \alpha^k d^k) = 0,$$

then we have both

$$\lim_{k\to\infty} \nabla f(x^k)^T d^k = 0, \quad \text{and} \quad \lim_{k\to\infty} \alpha^k \|d^k\| = 0 \tag{4.2}$$

Assumption 6. Let $\{x^k\}_K$ be a subsequence of feasible points convergent to a point x^*, such that $h^k = h \ \forall k \in K$. If

$$\lim_{k\in K, k\to\infty} \nabla f(x^k)^T d^k = 0,$$

we obtain

$$\nabla_{(h)} f(x^*)^T \big(x_{(h)} - x^*_{(h)}\big) \geq 0 \quad \forall\, x_{(h)} \in \mathcal{F}_h.$$

Proposition 2. Let $\{x^k\}$ be the sequence generated by IDA. Suppose that Assumptions 4 - 6 are satisfied. Then $\{x^k\}$ admits limit points and each limit point is a critical point for the problem (4.1).

See Proposition 1 from [16] for the complete proof. In practice, the above three assumptions are related to one or more steps of Algorithm 2 and require it to have some specific behaviors:

- Assumption 4 is satisfied by defining a suitable decomposition scheme (Step 3). In particular, it requires that each block h is periodically taken into account. In the proofs below, this is one of the ingredients that ensures that the discussion made on a single block might be extended to all the others;

- Assumption 5 is satisfied by defining a suitable line search mapping (Step 5) and a suitable descent direction (Step 4). This assumption ensures that the chosen combination between line search and direction is able to force directional derivative goes to zero. Moreover, it also guarantees that the distance between two consecutive iterates goes to zero. This is the second ingredient for extending a single-block discussion to the multi-block case;

- Assumption 6 is satisfied by defining a suitable descent direction (Step 4) and exploiting Assumption 5. This assumption requires that the sequence $\{d^k\}$ is gradient-related [10]. The properties of the direction are needed to obtain this condition.

4.3 The Nonmonotone Extended Framework

The first important point to notice, taking a closer look at the proof of Proposition 1 from [16], is that the monotonicity requirement $f(x^{k+1}) \leq f(x^k)$ is not really needed. In fact, the rest of the proof does only exploit conditions contained in (4.2) without a precise interest in how they are obtained. For this reason we might think of relaxing Assumption 5 and use instead Assumption 7 below. Thanks to this relaxation, Algorithm 2 might be extended to the use of nonmonotone line search mappings. Note that the following assumption permits that the sequence of the objective function values may be nonmonotone, but ensures that the points generated by the algorithm belong to the compact zero level set $\mathcal{L}_0 = \{x \in \mathcal{F} : f(x) \leq f(x^0)\}$.

Assumption 7. At every iteration k, the line search procedure computes a value α^k such that

$$f(x^k + \alpha^k d^k) \leq f(x^0).$$

Furthermore, if $\{x^k\}$ is a sequence of feasible points convergent to a point $\bar{x}$, then we have

$$\lim_{k\to\infty} \alpha^k \|d^k\| = 0 \tag{4.3}$$

and

$$\lim_{k\to\infty} \nabla f(x^k)^T d^k = 0. \tag{4.4}$$

The new convergence result is stated in Theorem 2.

Theorem 2. Let $\{x^k\}$ be the sequence generated by Algorithm 2. Suppose that Assumptions 4, 6 and 7 are satisfied. Then $\{x^k\}$ admits limit points that belong to $\mathcal{L}_0$ and each limit point is a critical point for problem (4.1).

Proof. The sequence $\{x^k\}$ of generated points is such that, for all k, $x^k \in \mathcal{F}$ since α^k and d^k are obtained by ensuring that the updated point is feasible (Step 4 and Step 5 of Algorithm 2). $\mathcal{F}$ is compact, thus $\{x^k\}$ admits limit points. Let x^* be a limit point of $\{x^k\}$, i.e., there exists an infinite subset $K \subset \mathbb{N}$ such that

$$\lim_{k\in K, k\to\infty} x^k = x^*.$$

By Assumption 7 we have that $\{x^k\} \subset \mathcal{L}_0$ and we obtain that

$$\lim_{k\in K, k\to\infty} \nabla f(x^k)^T d^k = 0, \quad \text{and} \quad \lim_{k\in K, k\to\infty} \alpha^k \|d^k\| = \|x^{k+1} - x^k\| = 0.$$

By induction, for every $l \in \mathbb{N}$, we can write

$$\lim_{k\in K, k\to\infty} x^{k+l} = x^*, \tag{4.5}$$

$$\lim_{k\in K, k\to\infty} \nabla f(x^{k+l})^T d^{k+l} = 0. \tag{4.6}$$

From Assumption 4 we have that for all $h \in \{1, \dots, L\}$ there exists an index $0 \leq j(k) \leq M$ such that $h^{k+j(k)} = h$. Thus, together with (4.5) and (4.6), we obtain that

$$\lim_{k\in K, k\to\infty} x^{k+j(k)} = x^*,$$

$$\lim_{k\in K, k\to\infty} \nabla f(x^{k+j(k)})^T d^{k+j(k)} = 0.$$

In conclusion, from the above two limits and Assumption 6, we have that

$$\nabla_{(h)} f(x^*)^T \left(x_{(h)} - x^*_{(h)}\right) \geq 0 \quad \forall x_{(h)} \in \mathcal{F}_h.$$

This inequality holds for every $h \in \{1, \dots, L\}$ and hence the proposition is proved. □

In the remaining of this section, we are going to prove that different nonmonotone line search mappings (Subsection 4.3.1) and direction choices (Subsection 4.3.2) do satisfy Assumptions 6 and 7. Note that in [16], nonmonotone line searches were not addressed and the direction defined by the Gauss-Southwell rule was not proved to obtain convergence. As it will be clearer in the following, Theorem 2 proves convergence for some combination of directions and line searches: Gauss-Southwell-rule-based & (nonmonotone) Quadratic, Frank-Wolfe & (nonmonotone) Quadratic, projected gradient & (nonmonotone) Quadratic and projected gradient & (nonmonotone) Armijo. All the four combinations need to prove Lemma 4 below and to satisfy Assumption 6 and 7 to obtain convergence. As for its monotone counterpart, Assumption 7 is not only employed in the final proof (of Theorem 2), but it is also required to satisfy Assumption 6.

4.3.1 Nonmonotone Line search mappings

In this subsection we are going to present two nonmonotone line search mappings that, together with a suitable direction, might be employed to satisfy Assumption 7: Armijo (see [5], Algorithm 3) and Quadratic (see [26], Algorithm 4) line searches. Both line searches will be extended to accept a nonmonotone step thanks to the original strategy introduced in [55]. In order to guarantee theoretical properties by either Armijo or Quadratic line search we need to ensure that the sequence of descent directions is uniformly bounded in $\mathcal{F}$ (Assumption 8 (b) or Assumption 9 (b)). Moreover, to obtain global convergence, as usual in the contexts of decomposition methods and nonmonotone strategies, we also need to ensure that the distance between consecutive iterates goes to 0. In turn, to obtain this, the Armijo line search requires Assumption 8 (a) and the Quadratic one requires instead the weaker Assumption 9 (a).

Given a point $x^k \in \mathcal{F}$ and a feasible direction d^k at x^k, we indicate by β^k the maximum feasible step length along d^k, i.e.,

$$x^k + \beta d^k \in \mathcal{F} \qquad \forall \beta \in [0, \beta^k] \qquad \text{and} \qquad x^k + \beta d^k \notin \mathcal{F} \quad \forall \beta > \beta^k.$$

As it will be clearer from Proposition 3 and 4, from the line search properties it is only possible to obtain (4.11) (or the corresponding (4.13)), i.e., the product between the directional derivative and β^k goes to 0. To obtain global convergence we need to ensure that the directional derivative alone is going to 0, but this is not straightforward as in the unconstrained case where β^k is always bounded from below. For this reason, also in this step of the global convergence proof, we need to exploit some additional characteristics of the direction.

Let $W > 0$ be the nonmonotone window, then we define

$$f^k_{max} := \max_{k-W \leq i \leq k} f(x^i) \tag{4.7}$$

as the largest function value among the last few iterations, where, formally, we set $x^{-1} := x^{-2} := \ldots := x^{-W} := x_0$ (or, alternatively, we can replace W by $W^k := \min\{k, W\}$). Furthermore, let us define

$$r(k) \leq k \text{ as the (largest) iteration index such that } f(x^{r(k)}) = f^k_{max} \text{ holds.} \tag{4.8}$$

Definition 1. A function $\sigma : \mathbb{R}_+ \to \mathbb{R}_+$ is called forcing function if for each sequence $\{t^k\}$ with $t^k \in \mathbb{R}_+$ we have that

$$\lim_{k\to\infty} \sigma(t^k) = 0 \quad \Rightarrow \quad \lim_{k\to\infty} t^k = 0.$$

Lemma 4. Suppose that $\mathcal{L}_0$ is compact. Let $\{x^k\}$ be a sequence of points such that $x^k \in \mathcal{F}$ and

$$f(x^{k+1}) \leq f^k_{max} - \sigma(\|x^{k+1} - x^k\|), \tag{4.9}$$

where σ is a forcing function, then we have

(a) $x^k \in \mathcal{L}_0$;

(b) $\{f^k_{max}\}$ and $\{f^k\}$ converge to the same limit;

(c) $\lim\limits_{k\to\infty} \|x^{k+1} - x^k\| = \lim\limits_{k\to\infty} \|\alpha^k d^k\| = 0.$

Thus, if the line search condition might be written as in (4.9), the limit (4.3) of Assumption 7 is obtained directly from Lemma 4 (see Lemma 2 of [53]). Note that this Lemma ensures that the sequence $\{x^k\}$ of points generated by the algorithm belongs to the compact set $\mathcal{L}_0$. In turn, this means that $\{x^k\}$ admits limit points and the function value in each of them is less or equal than $f(x^0)$.

Armijo Line Search

Let us first consider a nonmonotone Armijo line search mapping. Algorithm 3 acceptance condition is

```
Algorithm 3: nonmonotone Armijo Line search
Input: $x^k, d^k, \beta^k$(maximum feasible step length),
       $\lambda > 0, \delta \in (0,1), \gamma \in (0,1), W \in \mathbb{N}$
Output: $\alpha^k$
$\alpha = \min\{\beta^k, \lambda\}$
while $f(x^{k+1}) > f^k_{max} + \gamma\alpha\nabla f(x^k)^T d^k$ do
    $\alpha = \delta\alpha$
$\alpha^k = \alpha$
```

$$f(x^{k+1}) \leq f^k_{max} + \gamma\alpha^k \nabla f(x^k)^T d^k. \tag{4.10}$$

In order to write (4.10) as in (4.9) we need the following assumption.

Assumption 8. Let $\{d^k\}$ be a sequence of feasible search directions. Assume that

(a) $\exists\, c_1 > 0$ such that $\forall\, k$ we have $\nabla f(x^k)^T d^k \leq -c_1 \|d^k\|^2$,

(b) $\forall\, k$ we have $\|d^k\| \leq M$ for a given number $M > 0$.

Since (4.10) is a relaxed version of the original monotone Armijo condition, under Assumption 8 (a) we have that Algorithm 3 terminates finitely (see [5]).

Remark 1. From (4.10) and (a) of Assumption 8 we have that

$$f(x^{k+1}) \leq f^k_{max} - \gamma c_1 \alpha^k \|d^k\|^2 \leq f^k_{max} - \frac{\gamma c_1}{\lambda} \|\alpha^k d^k\|^2.$$

Thus, condition (4.9) of Lemma 4 holds if the forcing function is defined by $\sigma(t) = \frac{\gamma c_1}{\lambda} t^2$.

The following proposition shows that if we employ a Nonmonotone Armijo Line Search together with a search direction that satisfies Assumption 8 we obtain that the product between the directional derivative and the maximum feasible step length β^k goes to 0. Its proof is an adapted version the one of Proposition 4 from [16] (see Appendix 4.8).

Proposition 3. Let $\{d^k\}$ be a sequence of search directions satisfying Assumption 8. Let $\{x^k\}$ be a sequence of points generated by equation

$$x^{k+1} = x^k + \alpha^k d^k,$$

where α^k is obtained by the Nonmonotone Armijo Line Search (Algorithm 3). Then, if $\{x^k\}$ converges to $\bar{x}$, we have

$$\lim_{k \to \infty} \beta^k \nabla f(x^k)^T d^k = 0. \tag{4.11}$$

Quadratic Line Search

Now we take into account a nonmonotone Quadratic line search mapping.

Algorithm 4: nonmonotone Quadratic Line search

```
Input: $x^k, d^k, \beta^k$(maximum feasible step length),
       $\lambda > 0, \delta \in (0,1), \gamma \in (0,1), W \in \mathbb{N}$
Output: $\alpha^k$
$\alpha = \min\{\beta^k, \lambda\}$
while $f(x^{k+1}) > f^k_{max} - \gamma \|\alpha d^k\|^2$ do
    $\alpha = \delta\alpha$
$\alpha^k = \alpha$
```

Algorithm 4 acceptance condition is

$$f(x^{k+1}) \leq f^k_{max} - \gamma \|\alpha^k d^k\|^2. \tag{4.12}$$

In this case, we need something weaker than Assumption 8 to write (4.12) as in (4.9).

Assumption 9. Let $\{d^k\}$ be a sequence of feasible search directions. Assume that

(a) $\forall\, k$ we have $\nabla f(x^k)^T d^k < 0$;

(b) $\forall\, k$ we have $\|d^k\| \leq M$ for a given number $M > 0$.

Since (4.12) is a relaxed version of the original Quadratic monotone condition, under Assumption 9 (a) we have that Algorithm 4 terminates finitely (see Proposition 4.1 from [56]).

Remark 2. Note that condition (4.9) of Lemma 4 if the forcing function is defined by $\sigma(t) = \gamma t^2$.

The following proposition shows that if we employ a Nonmonotone Quadratic Line Search together with a search direction that satisfies Assumption 9 we obtain that the product between the directional derivative and the maximum feasible step length β^k goes to 0. Its proof of the following proposition is similar to the one of Proposition 5 from [16] (see Appendix 4.9).

Proposition 4. Let $\{d^k\}$ be a sequence of search directions satisfying Assumption 9. Let $\{x^k\}$ be a sequence of points generated by equation

$$x^{k+1} = x^k + \alpha^k d^k,$$

where α^k is obtained by the Nonmonotone Quadratic Line Search (Algorithm 4). Then, if $\{x^k\}$ converges to $\bar{x}$, we have

$$\lim_{k\to\infty} \beta^k \nabla f(x^k)^T d^k = 0. \tag{4.13}$$

4.3.2 Suitable Direction Choices

In Proposition 5, 6 and 7 we will respectively show that projected gradient direction, Frank-Wolfe direction and the sparse direction based on the Gauss-Southwell rule (in the case of single equality and box-constrained problems) satisfy Assumptions 6 and 7. Thus, from Assumption 7 we have that the direction should be gradient-related. While to satisfy Assumption 6, the direction needs to be combined with a line search mapping to first obtain Proposition 3 (or 4). In turn, the hypothesis of Proposition 3 (or 4) is Assumption 8 (or Assumption 9), which again needs to be satisfied exploiting the definition of the direction. We will show that Frank-Wolfe and the Gauss-Southwell-rule-based directions satisfy Assumption 9 (respectively in Proposition 6 and 7), while projected gradient direction satisfies both Assumption 8 and Assumption 9 (in Proposition 5).

As it will be clearer from the definitions, both projected gradient (4.14) and Frank-Wolfe (4.15) directions are always feasible. Furthermore, the maximum feasible step length β^k is always bounded from below ($\beta^k \geq 1$). Thus from Proposition 3 (or 4) is straightforward to obtain Assumption 7. On the other hand, for the Gauss-Southwell-rule-based direction (4.17)-(4.18) Assumption 7 is not that easy to be obtained. In this work we prove that if $\mathcal{F}_h$ has a certain structure we might exploit the property of the Gauss-Southwell rule in order to obtain Assumption 7. In the following we will implicitly assume that, if $\nabla f(x^k)^T d^k \geq 0$, then we will consider a null step along the search direction, that is $\alpha^k = 0$.

Projected Gradient Direction

Given $h^k \in \{1, \dots, L\}$ the selected block at iteration k, the projected gradient direction is defined by

$$d^k_{(h)} = \begin{cases} P_{\mathcal{F}_h}[x^k_{(h)} - \nabla_{(h)} f(x^k)] - x^k_{(h)}, & h^k = h, \\ 0, & h^k \neq h, \end{cases} \tag{4.14}$$

where $P_{\mathcal{F}_h}[\cdot]$ is the projection operator on the set $\mathcal{F}_h$. From the convexity of $\mathcal{F}$ it follows that $x^k + d^k \in \mathcal{F}$, so that we have that the maximum feasible step length β^k is greater or equal to 1. The proof of the following proposition is an adapted version of the one from Proposition 4 of [16] (see Appendix 4.10).

Proposition 5. Let $\{d^k\}$ be a sequence of search directions defined as in (4.14). Let $\{\alpha^k\}$ be a sequence of step lengths defined by an Armijo (or Quadratic) Line search. Then, Assumptions 6 and 7 are satisfied.

Frank-Wolfe Direction

Given $h^k \in \{1, \dots, L\}$ the selected block at iteration k, the Frank-Wolfe direction might be defined by

$$d^k_{(h)} = \begin{cases} \hat{x}^k_{(h)} - x^k_{(h)}, & h^k = h, \\ 0, & h^k \neq h, \end{cases} \tag{4.15}$$

$$\text{where } \hat{x}^k_{(h)} \in \arg \min_{x_{(h)} \in \mathcal{F}_h(x)} \nabla_{(h)} f(x^k)^T x_{(h)}.$$

Again, from the convexity of $\mathcal{F}$ it follows that $x^k + d^k \in \mathcal{F}$, so that we have that the maximum feasible step length β^k is greater or equal to 1. The proof of the following proposition is an adapted version of the one from Proposition 5 of [16] (see Appendix 4.11).

Proposition 6. Let $\{d^k\}$ be a sequence of search directions defined as in (4.15). Let $\{\alpha^k\}$ be a sequence of step lengths defined by a Quadratic Line search. Then, Assumptions 6 and 7 are satisfied.

Gauss-Southwell-rule-based Direction

In this subsection, our analysis is limited to the case in which the factor sets $\mathcal{F}_h$ of problem (4.1) are defined as follows

$$\mathcal{F}_h = \left\{ x_{(h)} \in \mathbb{R}^{n_h} : a^T_{(h)} x_{(h)} = b_{(h)},\ l_{(h)} \leq x_{(h)} \leq u_{(h)} \right\}, \tag{4.16}$$

where we assume that $a_{(h),i} \neq 0$ for $i \in \{1, \dots, n_h\}$. There are many problems for which $\mathcal{F}_h$ has the structure (4.16), e.g., Network Equilibrium problems (NE) [29], training of Support Vector Machines (SVM) [73,74], portfolio selection problems [66] and optimal control problems [6].

In the context of (4.16), a common approach in the literature (i.e., [29,72, 74]) has been that of selecting only 2 nonzero components of each block. This idea, suggested by the structure of the problem, is very well suited for large-scale problems, in which each step is reduced to a very simple operation. In order to select the two components it is usually employed a Gauss-Southwell rule which picks the couple of components that mostly violates optimality.

We first recall the following index sets definitions. Given a point $x \in \mathcal{F}$

and an index $h \in \{1, \dots, L\}$, with $x_{(h),i}$ the i-th component of $x_{(h)}$, we define

$$\begin{aligned}
&L_h(x) = \left\{i \in \{1, \dots, n_h\} : x_{(h),i} = l_{(h),i}\right\}, \\
&L_h^-(x) = \left\{i \in L_h(x) : a_{(h),i} < 0\right\}, \\
&L_h^+(x) = \left\{i \in L_h(x) : a_{(h),i} > 0\right\}, \\
&U_h(x) = \left\{i \in \{1, \dots, n_h\} : x_{(h),i} = u_{(h),i}\right\}, \\
&U_h^-(x) = \left\{i \in U_h(x) : a_{(h),i} < 0\right\}, \\
&U_h^+(x) = \left\{i \in U_h(x) : a_{(h),i} > 0\right\}, \\
&R_h(x) = L_h^+(x) \cup U_h^-(x) \cup \left\{i \in \{1, \dots, n_h\} : l_{(h),i} < x_{(h),i} < u_{(h),i}\right\}, \\
&S_h(x) = L_h^-(x) \cup U_h^+(x) \cup \left\{i \in \{1, \dots, n_h\} : l_{(h),i} < x_{(h),i} < u_{(h),i}\right\}.
\end{aligned}$$

Given a point $x^k \in \mathcal{F}$ we also define

$$\begin{aligned}
I_h(x^k) &= \left\{ i \in \{1, \dots, n_h\} : i \in \arg \max_{i \in R_h(x^k)} -\frac{\nabla_{(h),i} f(x^k)}{a_{(h),i}} \right\}, \\
J_h(x^k) &= \left\{ j \in \{1, \dots, n_h\} : j \in \arg \min_{j \in S_h(x^k)} -\frac{\nabla_{(h),j} f(x^k)}{a_{(h),j}} \right\}.
\end{aligned}$$

Now, given a selected block $h \in \{1, \dots, L\}$ and a component of this block $p \in \{1, \dots, n_h\}$, we define

$$d_{(h),p}^{i,j} = \begin{cases} \frac{1}{a_{(h),p}}, & p = i, \\ -\frac{1}{a_{(h),p}}, & p = j, \\ 0, & \text{otherwise.} \end{cases} \tag{4.17}$$

Thus, the direction at iteration k, is

$$d_{(h)}^k = \begin{cases} d_{(h)}^{i_h^k, j_h^k}, & h = h^k, \\ 0, & \text{otherwise,} \end{cases} \tag{4.18}$$

where $i_h^k \in I_h(x^k)$, $j_h^k \in J_h(x^k)$. Given an index $h \in \{1, \dots, L\}$ and $\bar{x}_{(h)} \in \mathcal{F}_h$, we also define $D_h(\bar{x}_{(h)})$ as the set of feasible directions at $\bar{x}$ with respect to $\mathcal{F}_h$ and $D_h^{RS}(\bar{x}_{(h)})$ the set

$$D_h^{RS}(\bar{x}_{(h)}) = \bigcup_{\substack{i \in R_h(\bar{x}) \\ j \in S_h(\bar{x}) \\ i \neq j}} d_{(h)}^{i,j}.$$

Let us first recall Proposition 6 and Proposition 7 from [16] and Proposition 3.3 from [74]. In the following they are called Lemma 5, 6 and 7.

Lemma 5. Let $\{x^k\}$ be a sequence of feasible points for problem (4.1), with factor sets $\mathcal{F}_h$ of the form (4.16), convergent to a point $\bar{x}$, and let $h \in \{1, ..., L\}$. Then for sufficiently large values of k we have

$$R_h(\bar{x}) \subseteq R_h(x^k) \quad S_h(\bar{x}) \subseteq S_h(x^k).$$

Lemma 6. Let x be a feasible point of problem (4.1), with factor sets $\mathcal{F}_h$ of the form (4.16) and let $h \in \{1, \ldots, L\}$. Then

$$\nabla_{(h)} f(x)^T \big(y_{(h)} - x_{(h)}\big) \geq 0 \quad \forall\, y_{(h)} \in \mathcal{F}_h,$$

if and only if

$$\max_{i \in R_h(x)} \left\{ -\frac{\nabla_{(h),i} f(x)}{a_{(h),i}} \right\} \leq \min_{j \in S_h(x)} \left\{ -\frac{\nabla_{(h),j} f(x)}{a_{(h),j}} \right\}.$$

Lemma 7. Given $h \in \{1, \ldots, L\}$ and $\bar{x}_{(h)} \in \mathcal{F}_h$, we have

$$\text{cone}\{D_h^{RS}(\bar{x}_{(h)})\} = D_h(\bar{x}_{(h)}),$$

where, given Y a set of m vectors ($y^i \in \mathbb{R}^n$), we define

$$cone(Y) := \{y \in \mathbb{R}^n : y = \sum_{l=1}^{m} \mu_l y_l, \mu_l \geq 0, l = 1, \ldots, m\}.$$

In the following, we will use Lemma 6 and 7 to prove that Assumptions 6 and 7 are satisfied. Note that the following proof differs from the one given in [29] since it does not only consider simplexes, but box-constrained problems, and does also differ from the one given in [74] since it does not consider exact line searches, but inexact ones.

Proposition 7. Let $\{d^k\}$ be a sequence of search directions defined as in (4.18). Let $\{\alpha^k\}$ be a sequence of step lengths defined by a Quadratic Line search. Then, Assumption 6 and 7 are satisfied.

Proof. We assume that x^k is not a stationary point. Thus, from Lemma 6 and definition (4.18) we have that

$$\nabla f(x^k)^T d^k = \frac{\nabla_{(h),i_h^k} f(x^k)}{a_{(h),i_h^k}} - \frac{\nabla_{(h),j_h^k} f(x^k)}{a_{(h),j_h^k}} < 0.$$

Again by definition (4.18) we have that $\|d^k\| = \sqrt{2}\ \ \forall k$. Thus, Assumption 9 is proven.

Now let us prove Assumptions 7. Let $\{x^k\}$ be a sequence convergent to $\bar{x}$. By contradiction assume that there exists a convergent subsequence K such that

$$\lim_{k\in K, k\to\infty} \nabla f(x^k)^T d^k = -\mu < 0,$$

which means that there exist at least a block $\hat{h}$ and at least a couple of components $\bar{i} \in R_{\hat{h}}(\bar{x})$ and $\bar{j} \in S_{\hat{h}}(\bar{x})$ for which

$$\frac{1}{a_{(\hat{h}),\bar{i}}} \cdot \frac{\partial f(\bar{x})}{\partial x_{(\hat{h}),\bar{i}}} < \frac{1}{a_{(\hat{h}),\bar{j}}} \cdot \frac{\partial f(\bar{x})}{\partial x_{(\hat{h}),\bar{j}}}. \tag{4.19}$$

By Step 3 of Algorithm 2 we have that $h^k \in \{1, \ldots, L\}$. For each $k \in K$, let $\Gamma_1^k, \ldots \Gamma_L^k$ be defined as follows

$$\Gamma_h^k = \left\{ i \in \{0, \ldots, (2n+1)M\} : h^{k+i} = h, k \in K \right\}.$$

From Assumption 4 we have that every M iterations each block is considered at least once. Every $2M$ iterations each block is considered at least 2 times, thus, repeating the argument till $2n+1$, we have that every $(2n+1)M$ iterations each block is considered at least $(2n+1)$ times. Thus, together with the fact that $|\Gamma_1^k| + \cdots + |\Gamma_L^k| = (2n+1)M$, we have that each of those sets contains at least $2n+1$ indexes and in particular $|\Gamma_{\hat{h}}^k| \geq 2n+1$. Now we have 2 cases:
Case (I) If there exists a set of constants $\{M_1, \ldots, M_L\}$ such that $\forall\ k \in K$ there exists a limited $0 < m_h(k) < M_h$ such that

$$h^{k+m_h(k)} = \hat{h} \text{ and } \alpha^{k+m_h(k)} < \beta^{k+m_h(k)}. \tag{4.20}$$

Case (II) If this set does not exist, at least one of those constants does not exist, w.l.o.g. assume that $M_{\hat{h}}$ does not exist, which means that

$$h^{k+m} = \hat{h} \text{ and } \alpha^{k+m} = \beta^{k+m}, \quad \forall k \in K,\ \forall\ m \in \Gamma_{\hat{h}}^k.$$

Let us start with Case (I). From (4.20) we might exploit a failure in the line search procedure, which means that for block $\hat{h}$ we have

$$\begin{aligned} f(x^{k+m_{\hat{h}}(k)} + \frac{\alpha^{k+m_{\hat{h}}(k)}}{\delta} d^{k+m_{\hat{h}}(k)}) &> f_{max}^{k+m_{\hat{h}}(k)} - \gamma \| \frac{\alpha^{k+m_{\hat{h}}(k)}}{\delta} d^{k+m_{\hat{h}}(k)} \|^2 \\ &\geq f(x^{k+m_{\hat{h}}(k)}) - \gamma \| \frac{\alpha^{k+m_{\hat{h}}(k)}}{\delta} d^{k+m_{\hat{h}}(k)} \|^2. \end{aligned}$$

From the Mean Value Theorem, applied on the above inequality, we obtain that

$$\nabla f(z^{k+m_{\hat h}(k)})^T d^{k+m_{\hat h}(k)} > -\frac{\gamma}{\delta}\alpha^{k+m_{\hat h}(k)}\|d^{k+m_{\hat h}(k)}\|^2, \tag{4.21}$$

where $z^{k+m_{\hat h}(k)} = x^{k+m_{\hat h}(k)} + \theta^{k+m_{\hat h}(k)}\frac{\alpha^{k+m_{\hat h}(k)}}{\delta}d^{k+m_{\hat h}(k)}$, with $\theta^{k+m_{\hat h}(k)} \in [0,1]$. Since $m_{\hat h}(k) < M_{\hat h}$ we have that

$$\lim_{k\to\infty, k\in K} x^{k+m_{\hat h}(k)} = \bar x. \tag{4.22}$$

Thus, from Lemma 4 we also get that $\{z^{k+m_{\hat h}(k)}\}$ converges to $\bar x$. Now, since $i^{k+m_{\hat h}(k)}$ and $j^{k+m_{\hat h}(k)}$ belong to a finite set $\{1,\dots,L\}$ we can extract a further subset K, that we relabel again K, such that

$$i^{k+m_{\hat h}(k)} = \hat\imath \text{ and } j^{k+m_{\hat h}(k)} = \hat\jmath \text{ and } h^{k+m_{\hat h}(k)} = \hat h \quad \forall k \in K. \tag{4.23}$$

Then

$$\lim_{k\to\infty,k\in K} \nabla f(z^{k+m_{\hat h}(k)})^T d^{k+m_{\hat h}(k)} = \frac{1}{a_{(\hat h),\hat\imath}} \cdot \frac{\partial f(\bar x)}{\partial x_{(\hat h),\hat\imath}} - \frac{1}{a_{(\hat h),\hat\jmath}} \cdot \frac{\partial f(\bar x)}{\partial x_{(\hat h),\hat\jmath}} = \nu.$$

Now, from (4.21), (b) of Assumption 9 and (c) of Lemma 4, we get that that

$$\lim_{k\to\infty,k\in K} \frac{\gamma}{\delta}\alpha^{k+m_{\hat h}(k)}\|d^{k+m_{\hat h}(k)}\|^2 = 0 \le \nu. \tag{4.24}$$

From Lemma 5 and (4.22) we have that

$$R_{\hat h}(\bar x) \subseteq R_{\hat h}(x^k), R_{\hat h}(\bar x) \subseteq R_{\hat h}(x^{k+m_{\hat h}(k)}) \text{ and}$$
$$S_{\hat h}(\bar x) \subseteq S_{\hat h}(x^k), S_{\hat h}(\bar x) \subseteq S_{\hat h}(x^{k+m_{\hat h}(k)}),$$

thus, $\forall k \in K$ sufficiently large, we have that

$$\bar\imath \in R_{\hat h}(x^{k+m_{\hat h}(k)}) \text{ and } \bar\jmath \in S_{\hat h}(x^{k+m_{\hat h}(k)}),$$

which means that

$$\frac{1}{a_{(\hat h),\hat\imath}} \cdot \frac{\partial f(x^{k+m_{\hat h}(k)})}{\partial x_{(\hat h),\hat\imath}} \le \frac{1}{a_{(\hat h),\bar\imath}} \cdot \frac{\partial f(x^{k+m_{\hat h}(k)})}{\partial x_{(\hat h),\bar\imath}}$$

and

$$\frac{1}{a_{(\hat h),\hat\jmath}} \cdot \frac{\partial f(x^{k+m_{\hat h}(k)})}{\partial x_{(\hat h),\hat\jmath}} \ge \frac{1}{a_{(\hat h),\bar\jmath}} \cdot \frac{\partial f(x^{k+m_{\hat h}(k)})}{\partial x_{(\hat h),\bar\jmath}},$$

thus we can write

$$\frac{1}{a_{(\hat{h}),\bar{\imath}}} \cdot \frac{\partial f(x^{k+m_{\hat{h}}(k)})}{\partial x_{(\hat{h}),\bar{\imath}}} - \frac{1}{a_{(\hat{h}),\bar{\jmath}}} \cdot \frac{\partial f(x^{k+m_{\hat{h}}(k)})}{\partial x_{(\hat{h}),\bar{\jmath}}} \geq$$
$$\frac{1}{a_{(\hat{h}),\hat{\imath}}} \cdot \frac{\partial f(x^{k+m_{\hat{h}}(k)})}{\partial x_{(\hat{h}),\hat{\imath}}} - \frac{1}{a_{(\hat{h}),\hat{\jmath}}} \cdot \frac{\partial f(x^{k+m_{\hat{h}}(k)})}{\partial x_{(\hat{h}),\hat{\jmath}}},$$

from which, takings limits and using (4.24), we obtain that

$$\frac{1}{a_{(\hat{h}),\bar{\imath}}} \cdot \frac{\partial f(\bar{x})}{\partial x_{(\hat{h}),\bar{\imath}}} - \frac{1}{a_{(\hat{h}),\bar{\jmath}}} \cdot \frac{\partial f(\bar{x})}{\partial x_{(\hat{h}),\bar{\jmath}}} \geq$$
$$\frac{1}{a_{(\hat{h}),\hat{\imath}}} \cdot \frac{\partial f(\bar{x})}{\partial x_{(\hat{h}),\hat{\imath}}} - \frac{1}{a_{(\hat{h}),\hat{\jmath}}} \cdot \frac{\partial f(\bar{x})}{\partial x_{(\hat{h}),\hat{\jmath}}} \geq 0,$$

which is absurd by (4.19).

Let us now consider Case (II). For all $m \in \Gamma^k_{\hat{h}}$ we have that at least one of the possible two cases holds:

$$i^{k+m}_{\hat{h}} \in R_{\hat{h}}(x^{k+m}) \qquad i^{k+m}_{\hat{h}} \notin R_{\hat{h}}(x^{k+m+1}) \tag{4.25}$$
$$j^{k+m}_{\hat{h}} \in S_{\hat{h}}(x^{k+m}) \qquad j^{k+m}_{\hat{h}} \notin S_{\hat{h}}(x^{k+m+1}), \tag{4.26}$$

where $i^{k+m}_{\hat{h}} \in I_{\hat{h}}(x^{k+m})$, $j^{k+m}_{\hat{h}} \in J_{\hat{h}}(x^{k+m})$. Now we define 2 sets $\Gamma^{k'}_{\hat{h}}$ and $\Gamma^{k''}_{\hat{h}}$ in which (4.25) and (4.26) are respectively satisfied, and $\Gamma^{k'}_{\hat{h}} \cup \Gamma^{k''}_{\hat{h}} = \Gamma^k_{\hat{h}}$. Thus, one of them contains more than n indexes, and, w.l.o.g, we assume that $|\Gamma^{k'}_{\hat{h}}| > n$. Thus, there exists $\hat{\imath} \in \{1, \ldots, n\}$, and $l(k), m(k)$ such that

$$k \leq l(k) < m(k) \leq k + 2nM$$

and

$$i^{l(k)} = i^{m(k)} = \hat{\imath} \text{ and } h^{l(k)} = i^{m(k)} = \hat{h}.$$

Since K is infinite and n is limited, we can define a subset $K_1 \subseteq K$ such that the previous argument is valid on $\hat{\imath}$ for each $k_i \in K_1$, thus we have

$$i^{l(k_i)} = i^{m(k_i)} = \hat{\imath} \text{ and } h^{l(k_i)} = h^{m(k_i)} = \hat{h}$$

with

$$k_i \leq l(k_i) < m(k_i) \leq k_i + 2nM.$$

Thus, $\forall\ k_i \in K_1$, from (4.17) we have that there exists an index $p(k_i)$, with $k_i \leq l(k_i) < p(k_i) < m(k_i) \leq k_i + 2nM$ and $h^{p(k_i)} = \hat{h}$, in which $x_{(\hat{h}),\hat{\imath}}$ is moved away from its previous bound, between iterations $l(k_i)$ and $m(k_i)$, i.e.,

$$\hat{\imath} \notin R_{\hat{h}}(x^{p(k_i)}) \quad \hat{\imath} \in R_{\hat{h}}(x^{p(k_i)+1}). \tag{4.27}$$

Thus, from the Gauss-Southwell rule we have that

$$\frac{1}{a_{(\hat{h}),\hat{\imath}}} \cdot \frac{\partial f(x^{p(k_i)})}{\partial x_{(\hat{h}),\hat{\imath}}} \geq \frac{1}{a_{(\hat{h}),j}} \cdot \frac{\partial f(x^{p(k_i)})}{\partial x_{(\hat{h}),j}} \;\; \forall j \in S_{\hat{h}}(x^{p(k_i)}), \tag{4.28}$$

and also

$$\frac{1}{a_{(\hat{h}),\hat{\imath}}} \cdot \frac{\partial f(x^{l(k_i)})}{\partial x_{(\hat{h}),\hat{\imath}}} \leq \frac{1}{a_{(\hat{h}),i}} \cdot \frac{\partial f(x^{l(k_i)})}{\partial x_{(\hat{h}),i}} \;\; \forall i \in R_{\hat{h}}(x^{l(k_i)}). \tag{4.29}$$

As $p(k_i) - k_i \leq 2nM$ and $l(k_i) - k_i \leq 2nM$, we can write

$$\lim_{k_i \in K_1, k \to \infty} x^{p(k_i)} = \bar{x} = \lim_{k_i \in K_1, k \to \infty} x^{l(k_i)}. \tag{4.30}$$

From (4.28), (4.29) and (4.30) we obtain that

$$\begin{aligned}\max_{i \in R_{\hat{h}}(\bar{x})} \left\{ -\frac{1}{a_{(\hat{h}),i}} \cdot \frac{\partial f(x^*)}{\partial x_{(\hat{h}),i}} \right\} &= \min_{j \in S_{\hat{h}}(\bar{x})} \left\{ -\frac{1}{a_{(\hat{h}),j}} \cdot \frac{\partial f(x^*)}{\partial x_{(\hat{h}),j}} \right\} \\ &= -\frac{1}{a_{(\hat{h}),\hat{\imath}}} \cdot \frac{\partial f(x^*)}{\partial x_{(\hat{h}),\hat{\imath}}},\end{aligned}$$

which contradicts (4.19) since $\bar{\imath} \in R_{\hat{h}}(\bar{x})$ and $\bar{\jmath} \in S_{\hat{h}}(\bar{x})$, proving Assumption 7.

Now let us prove Assumption 6. Let $\{x^k\}_K$ be a sequence convergent to x^* for which $h^k = h \;\; \forall k \in K$ and

$$\lim_{k \in K, k \to \infty} \nabla f(x^k)^T d^k = 0. \tag{4.31}$$

Assume by contradiction that $\exists \; \hat{x}_{(h)} \in \mathcal{F}_h$ such that

$$\nabla_{(h)} f(x^*_{(h)})(\hat{x}_{(h)} - x^*_{(h)}) < 0. \tag{4.32}$$

Let us define $\hat{d}_{(h)} := \hat{x}_{(h)} - x^*_{(h)}$. From Lemma 7, we have that

$$\hat{d}_{(h)} = \sum_{l=1}^{|D_h^{RS}(x^*_{(h)})|} \hat{d}_{(h),l}\mu_l,$$

where $\mu_l \geq 0$ and $\hat{d}_{(h),l} \in D_h^{RS}(x^*_{(h)})$. From $\mu_l \geq 0$ and (4.32) there must exists at least a direction $\hat{d}_{(h),\hat{l}}$ for which $\nabla_{(h)} f(x^k)^T \hat{d}_{(h),\hat{l}} < 0$. From the definition of Gauss-Southwell-rule-based direction (4.18) we have that

$$\nabla f(x^k)^T d^k = \nabla_{(h)} f(x^k)^T d^k_{(h)} \leq \nabla_{(h)} f(x^k)^T \hat{d}_{(h),\hat{l}} < 0,$$

which means that the sequence $\{\nabla f(x^k)^T d^k\}$ is bounded away from 0, which is absurd from (4.31). □

4.4 On some Numerical Issues

Let us consider a constrained problem in which the objective function is partially separable, i.e.,

$$\begin{aligned} &\min f(x) = \sum_{h=1}^{N} f_h(x) \\ &\text{s.t. } x \in \mathbb{X} \subseteq \mathbb{R}^n, \end{aligned} \tag{4.33}$$

where $f : \mathbb{X} \to \mathbb{R}$ and each individual $f_h(x)$ is a function depending on some of the components x_i, $i \in I_h$ for some index sets $I_h \subseteq \{1, \dots, n\}$, $h = 1, \dots, N$.

For a given working set $w \subseteq \{1, \dots, n\}$, we define an index set $H_w \subseteq \{1, \dots, N\}$ of function $f_h(x)$ dependent on one or more components x_i belonging to w, that is, $H_w = \{h \in \{1, \dots, N\} \mid w \cap I_h \neq \emptyset\}$. Given H_w, we can easily define the objective function part f_w related to $w \subseteq \{1, \dots, n\}$ as follows

$$f_w(x) = \sum_{h \in H_w} f_h(x). \tag{4.34}$$

In the same manner, we can define the complementary part $f_{\bar{w}}$ of f_w

$$f_{\bar{w}}(x) = \sum_{h \in \{1,\dots,N\} \setminus H_w} f_h(x) \tag{4.35}$$

such that

$$f(x) = f_{\bar{w}}(x) + f_w(x), \qquad \forall\, x \in X, \forall\, w \subseteq \{1, \dots, n\}\,. \tag{4.36}$$

In order to show how to take advantage of using $f_w(x)$ in place of f from both the computational and numerical points of view we assume that

- computing f_w is far less expensive than computing the whole f,
- given a suitable $w \subseteq \{1, \ldots, n\}$ we have

$$f(x) \gg f_w(x). \tag{4.37}$$

Those assumptions are often satisfied when facing large-scale problems, in fact, the working sets w are usually built to deal with dimensions which are much smaller than the originals.

4.4.1 The Monotone Numerical Trick

In this subsection we formalize and generalize a smart numerical trick that is part of the common knowledge in various specific fields (for instance see Remark 2 from [15]). We approach (4.33) using a decomposition framework together with a line search. That is, at each iteration k, the working set $w^k \subseteq \{1, \ldots, n\}$ is considered and the point x^k is updated as follows

$$x^{k+1} = x^k + \alpha^k d^k,$$

where $d^k \neq 0$ is a feasible direction and $d_i^k = 0$, $\forall\, i \notin w^k$, and α^k is the step along d^k obtained by means of a line search procedure. Now, from (4.34), (4.35) and (4.36), we have

$$f(x^{k+1}) = f_{\bar{w}^k}(x^{k+1}) + f_{w^k}(x^{k+1}).$$

From the definition of H_w and d^k, it holds that $f_{\bar{w}^k}(x^{k+1}) = f_{\bar{w}^k}(x^k)$, thus we have

$$\begin{aligned} f(x^{k+1}) &= f_{\bar{w}^k}(x^k) + f_{w^k}(x^{k+1}) = \\ &= f_{\bar{w}^k}(x^k) + f_{w^k}(x^k) + f_{w^k}(x^{k+1}) - f_{w^k}(x^k) = \\ &= f(x^k) + f_{w^k}(x^{k+1}) - f_{w^k}(x^k). \end{aligned} \tag{4.38}$$

It is easy to see that $f(x^{k+1})$ can be updated evaluating only f_{w^k}, more precisely, it is updated adding the reduction of f_{w^k} obtained by the line search. As f is much more expensive than f_{w^k}, a relevant computational saving can be obtained.

From a numerical point of view, since $f(x) \gg f_w(x)$ and consequently $f_{\bar{w}}(x) \gg f_w(x)$, the evaluation of the sufficient reduction performed by the line search could be fairly noisy. In particular, in later stages of the optimization, it might hold that

$$f(x^{k+1}) \approx f(x^k) \tag{4.39}$$

from a numerical point of view. The monotone numerical trick exploits the definition of f in (4.36). For example, using a quadratic line search rule as the following

$$f(x^{k+1}) \leq f(x^k) - \gamma \|\alpha^k d^k\|^2,$$

from (4.36) we have

$$f_{\bar{w}^k}(x^{k+1}) + f_{w^k}(x^{k+1}) \leq f_{\bar{w}^k}(x^k) + f_{w^k}(x^k) - \gamma \|\alpha^k d^k\|^2,$$

then, since $f_{\bar{w}^k}(x^{k+1}) = f_{\bar{w}^k}(x^k)$, we obtain

$$f_{\bar{w}^k}(x^k) + f_{w^k}(x^{k+1}) \leq f_{\bar{w}^k}(x^k) + f_{w^k}(x^k) - \gamma \|\alpha^k d^k\|^2$$

and finally

$$f_{w^k}(x^{k+1}) \leq f_{w^k}(x^k) - \gamma \|\alpha^k d^k\|^2. \tag{4.40}$$

Using (4.40), the noisy comparison between $f(x^{k+1})$ and $f(x^k)$ can be avoided. In the illustrative example reported in Section 4.5 we will show that the computation of the monotone line search step length is much more accurate than that of using f, as well as the computational savings obtained with f_{w^k}.

4.4.2 The Nonmonotone Numerical Trick

To the best of our knowledge, the nonmonotone numerical trick derived in this subsection was never proposed in the literature before. Let us now assume that the line search rule is nonmonotone. For instance, we employ the classical nonmonotone strategy from [55] on the quadratic line search

$$f(x^{k+1}) \leq f^k_{max} - \gamma \|\alpha^k d^k\|^2 \tag{4.41}$$

where f^k_{max} is defined in (4.7).

Conversely from what happens in the monotone case, since f^k_{max} is generally related to a past iteration, the term $f_{\bar{w}^k}(x^k)$ can't be deducted from both sides as in (4.40). However, removing $f(x^k)$ from both sides in (4.41), we obtain

$$f(x^{k+1}) - f(x^k) \leq f^k_{max} - f(x^k) - \gamma \|\alpha^k d^k\|^2$$

and from (4.38) we have

$$f_{w^k}(x^{k+1}) - f_{w^k}(x^k) \leq f^k_{max} - f(x^k) - \gamma\|\alpha^k d^k\|^2$$

and finally

$$f_{w^k}(x^{k+1}) \leq f_{w^k}(x^k) + \Delta^k_{max} - \gamma\|\alpha^k d^k\|^2, \tag{4.42}$$

where

$$\Delta^k_{max} = f^k_{max} - f(x^k) \tag{4.43}$$

is the maximum distance between $f(x^k)$ and any objective function value seen in the past W iterations. In case $W = 1$, we get back to the monotone case (4.40) and $\Delta^k_{max} = 0$.

In (4.42), the numerical raw comparison between $f(x^{k+1})$ and $f(x^k)$ is again avoided and the achieved computational benefit is the same as in the monotone case. However, the computation of Δ^k_{max} is critical from a numerical point of view, as it is a difference between f^k_{max} and $f(x^k)$ which are generally close since the nonmonotone window W is finite and assumption (4.37) holds. This will lead to a catastrophic cancellation due to the lack of significant digits in the finite real-value floating-point representation.

If the numerical issue of computing Δ^k_{max} is not addressed properly, the nonmonotone behavior may end after a few iterations, as it happens in practice. For this reason, the following nonmonotone numerical trick has been designed. We recall that $r(k)$ is the index of the nonmonotone window correspondent to f^k_{max} defined in (4.8). The term Δ^k_{max} is then obtained as follows

$$\begin{aligned}
\Delta^k_{max} &= f^k_{max} - f(x^k) = \\
&= f(x^{k-r(k)}) - f(x^k) = \\
&= f(x^{k-r(k)}) - f(x^{k-r(k)+1}) + f(x^{k-r(k)+1}) - f(x^k) = \\
&= f(x^{k-r(k)}) - f(x^{k-r(k)+1}) + f(x^{k-r(k)+1}) - \ldots + \\
&\quad - f(x^{k-1}) + f(x^{k-1}) - f(x^k) = \\
&= \sum_{i=1,\ldots,r(k)} f(x^{k-i}) - f(x^{k-i+1}).
\end{aligned}$$

Finally, since it holds that

$$f(x^k) - f(x^{k+1}) = f_{w^k}(x^k) - f_{w^k}(x^{k+1}),$$

from (4.38), we obtain

$$\begin{aligned}\Delta^k_{max} &= \sum_{i=1,\dots,r(k)} f(x^{k-i}) - f(x^{k-i+1}) = \\ &= \sum_{i=1,\dots,r(k)} f_{w^{k-i}}(x^{k-i}) - f_{w^{k-i}}(x^{k-i+1}).\end{aligned} \tag{4.44}$$

From (4.44), the computation of Δ^k_{max} needs only to store the reductions of subsequent line searches optimizations, exploiting the structure of f and the definition of the sets H_w, $w \subseteq \{1,\dots,n\}$. The computation of Δ^k_{max} is more robust than that of using f^k_{max} and $f(x^k)$. Moreover, the numerical consistency in terms of useful significant digits scales suitably in accordance with the order of magnitude of subsequent line search reductions, as Section 4.5 will show.

4.5 A Class of Large-Scale Problems

As an illustrative example, we show the numerical efficiency of the proposed nonmonotone algorithm on a set of large NE problems. Let $G = (N, A)$ be a graph, where N is the set of nodes and A is the set of arcs. Let $P \subseteq N \times N$ be the set of all Origin/Destination (O/D) pairs ($|P| = L$). For each $h \in P$, a travel demand $D_h > 0$ is defined. Each block $h \in \{1,\dots,L\}$ is here related to one O/D pair $h \in \{1,\dots,|P|\}$. Given such a network, the NE problem concerns the forecasting of the flow on each arc. We take into account the path-based formulation of the NE (see [29] for a complete description), where the problem structure is exactly the one from (4.1). In particular, indicated by x the vector of path flows, the objective function is a smooth, convex function defined by

$$f(x) = \sum_{a\in A} \int_0^{v_a(x)} s_a(t)\, dt,$$

where $v_a(x)$ is the total flow on arc a and $s_a(\cdot)$ is the arc-separable cost, and the factor sets are defined by

$$\mathcal{F}_h = \{x_{(h)} \in \mathbb{R}^{n_h} : \sum_{i\in\{1,\dots,n_h\}} x_{(h),i} = D_h,\ x_{(h)} \geq 0\}. \tag{4.45}$$

In (4.45), for each O/D pair $h \in \{1,\dots,L\}$, $x_{(h)} \in \mathbb{R}^{n_h}$ represents the vector of flows on the paths insisting on h. Note that the above formulation might be considered “virtual”, since the total number of paths is exponential in the dimension of the network. For this reason we define a restricted set $\mathcal{F}_h(x^k) \subseteq \mathcal{F}_h$, in which only paths with nonzero flow are considered. Then, variables are iteratively added by means of a column generation technique.

To solve the NE problem, Algorithm 2 was implemented by employing the Gauss-Southwell-rule-based direction with 2 nonzero components in Step 4 and a (nonmonotone) Quadratic line search in Step 5. The external decomposition (Step 3) simply iterates through the O/D pairs, which means that Assumption 4 holds with $M = L$. The column generation scheme is periodically applied to find the shortest paths to be added to the working set.

As regards the computation of the initial step length along the search direction, a commonly used choice [18] is the one obtained by minimizing the quadratic approximation of f along the direction. Taking into account that the direction has only 2 nonzero components and that the partial derivatives of f represent the costs of the corresponding paths, the closed-form solution of the above step length can be easily and efficiently calculated. This step length was showed to perform well in practice if employed directly, but to ensure global convergence a line search mapping is needed. In the monotone setting, it might happen that a sufficient decrement is not obtained employing this step, resulting in at least one reduction of it. This is the precise context for which nonmonotone strategies were designed: accepting as often as possible a good initial step without reducing it.

In this context we compare Algorithm 2 in which Step 5 is a nonmonotone Quadratic line search (Nonmonotone IDA, NIDA) with its monotone counterpart (IDA). Different executions are compared w.r.t. several values of the *relative gap*, a well-known measure of convergence in NE problems (see [29]). All the test problems used for the experiments are freely available at the web page `http://www.bgu.ac.il/~bargera/tntp/` and are described in Table 4.1. Both algorithms have been implemented in `C++`, the shortest path algorithm employed is Dijkstra from the C++ Boost libraries and parameters chosen for both the line searches are

$$\lambda = 1, \quad \delta = 0.5, \quad \gamma = 10^{-2}, \quad W = 10.$$

The two algorithms have been compared on the following measures: total number of iterations (Table 4.2) and total number of evaluations of the objective function (Table 4.3) w.r.t. several levels of the relative gap, from a rough equilibrium solution (10^{-4}) to a more accurate one (10^{-7}). From Table 4.2

Network	Label	# links	# nodes	# centroids	# O/D pairs
Sioux-Falls	SF	76	24	24	528
Winnipeg	W	2,535	1,067	154	4,345
Barcelona	B	2,522	1,020	110	7,922
Chicago-Sketch	CS	2,950	933	387	93,135
Berlin-Center	BC	28,376	12,981	865	49,688
Philadelphia	P	40,003	13,389	1,525	1,149,795
Chicago-Regional	CR	39,018	12,982	1,790	2,296,227
Sydney	S	75,379	33,113	3,264	3,340,619

Table 4.1: Network datasets details

Dataset	Algorithm	10^{-4}	10^{-5}	10^{-6}	10^{-7}
SF	monotone	2.69e+03	7.54e+03	2.23e+04	3.72e+04
	nonmonotone	1.98e+03	5.11e+03	1.12e+04	1.76e+04
W	monotone	2.56e+04	3.93e+04	159289	2.32e+05
	nonmonotone	1.07e+04	2.02e+04	6.89e+04	1.08e+05
B	monotone	3.87e+04	9.83e+04	279352	6.38e+05
	nonmonotone	1.06e+04	1.61e+04	4.68e+04	1.50e+05
CS	monotone	1.65e+05	2.50e+05	3.86e+05	6.72e+05
	nonmonotone	9.40e+04	1.23e+05	1.41e+05	2.27e+05
BC	monotone	8.63e+03	1.21e+04	1.90e+04	2.15e+04
	nonmonotone	5.04e+03	6.26e+03	8.86e+03	1.09e+04
P	monotone	1.08e+07	3.43e+07	9.88e+07	2.28e+08
	nonmonotone	4.03e+06	7.41e+06	1.75e+07	5.39e+07
CR	monotone	1.99e+07	3.53e+07	8.88e+07	1.36e+08
	nonmonotone	8.406e+06	1.47e+07	3.64e+07	6.01e+07
S	monotone	5.72e+06	8.57e+06	1.56e+07	2.40e+07
	nonmonotone	3.13e+06	3.42e+06	4.35e+06	6.72e+06

Table 4.2: Total number of iterations required by the monotone (IDA) and nonmonotone (NIDA) algorithm with respect to the relative gap.

and 4.3 it is possible to observe that the nonmonotone algorithm outperforms the original monotone version with respect to both number of iterations and function evaluations. In fact, NIDA obtains better performances than IDA for the whole optimization procedure. Moreover at a relative gap equal to 10^{-7}, the number of iterations (Table 4.2) of NIDA is always from twice to

Dataset	Algorithm	10^{-4}	10^{-5}	10^{-6}	10^{-7}
SF	monotone	2.82e+05	8.95e+05	2.76e+06	4.47e+06
	nonmonotone	1.04e+04	3.35e+04	7.25e+04	1.16e+05
W	monotone	6.69e+06	1.09e+07	4.90e+07	7.00e+07
	nonmonotone	1.67e+05	4.48e+05	1.69e+06	2.90e+06
B	monotone	1.66e+07	5.06e+07	1.61e+08	3.76e+08
	nonmonotone	5.82e+05	1.32e+06	6.27e+06	2.74e+07
CS	monotone	2.61e+07	4.89e+07	8.51e+07	1.54e+08
	nonmonotone	9.39e+05	1.84e+06	2.48e+06	5.58e+06
BC	monotone	2.03e+06	3.62e+06	6.41e+06	7.42e+06
	nonmonotone	1.26e+04	5.71e+04	7.86e+04	8.02e+04
P	monotone	3.24e+09	1.30e+10	3.62e+10	5.83e+10
	nonmonotone	8.60e+07	3.19e+08	1.12e+09	3.74e+09
CR	monotone	3.11e+09	6.48e+09	1.52e+10	1.93e+10
	nonmonotone	2.70e+07	1.35e+08	5.62e+08	9.48e+08
S	monotone	1.19e+09	2.46e+09	5.66e+09	8.76e+09
	nonmonotone	8.49e+06	2.10e+07	1.09e+08	3.20e+08

Table 4.3: Total number of function evaluations required by the monotone (IDA) and nonmonotone (NIDA) algorithm with respect to the relative gap.

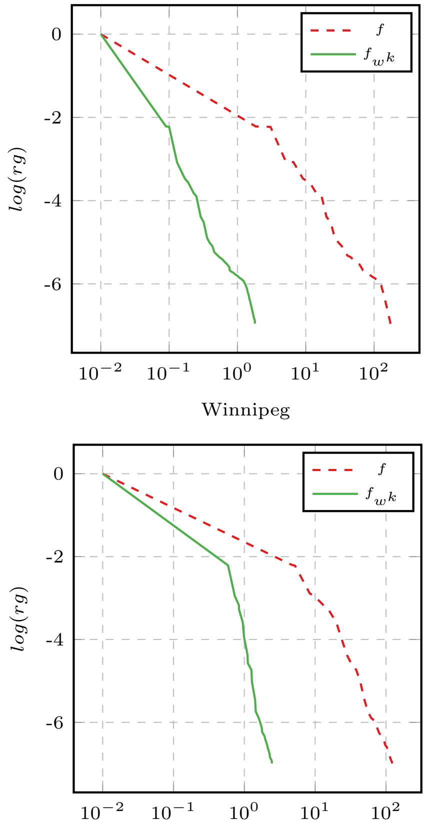
four times lower than IDA's.

The gain in terms of function evaluations (Table 4.3) is even grater, because the nonmonotone strategy is designed to relax the line search condition, accept more often the new step and, thus, reduce the number of function evaluations in each line search call.

Note that it was possible to preserve a remarkable gain also for higher accuracy only thanks to the numerical framework proposed in Section 4.4. Note that both IDA and NIDA employ a decomposition on the O/D pairs and iteratively equilibrates only two paths. This means that the working set w^k is made of only two variables. Thus, the portion of network edges affected by the equilibration is small enough to have that $f_{w^k} \ll f$ as in (4.37). Moreover, the equilibrium problem objective function is the sum of negative strictly increasing functions, one for each edge of the network (see [29]).

In the following we show the benefits of employing the numerical techniques discussed in Section 4.4 on two small-medium networks, Winnipeg and ChicagoSketch.

In Figure 4.1 the time needed to reach several levels of the *relative gap* using f_{w^k} and f is reported, where a logarithmic scale on the CPU time has been employed in order to properly compare f and f_{w^k}. From Figure 4.1, it is clear that using f_{w^k} leads to a relevant save of time. However, we need to show that the computation of f_{w^k} is at least as accurate as f from a numerical point of view. In order to compare the robustness of both f_{w^k} and f, we evaluate the numerical error with respect to the evaluation of the objective function using the quadruple precision floating-point representation (128-bit), while standard computations are in double precision (64-bit).

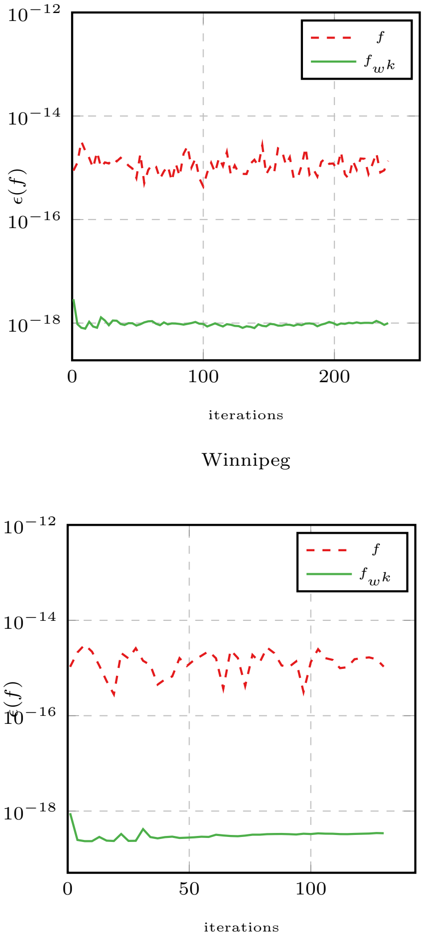


Winnipeg

ChicagoSketch

Figure 4.1: Computational comparison between the use of f and f_{w^k} in the line search procedure. CPU time (s).

Such reference value denoted with f^∞ is considered as the ground truth



Winnipeg

ChicagoSketch

Figure 4.2: Objective function evaluation error with f and f_{w^k} in the line search procedure through iterations.

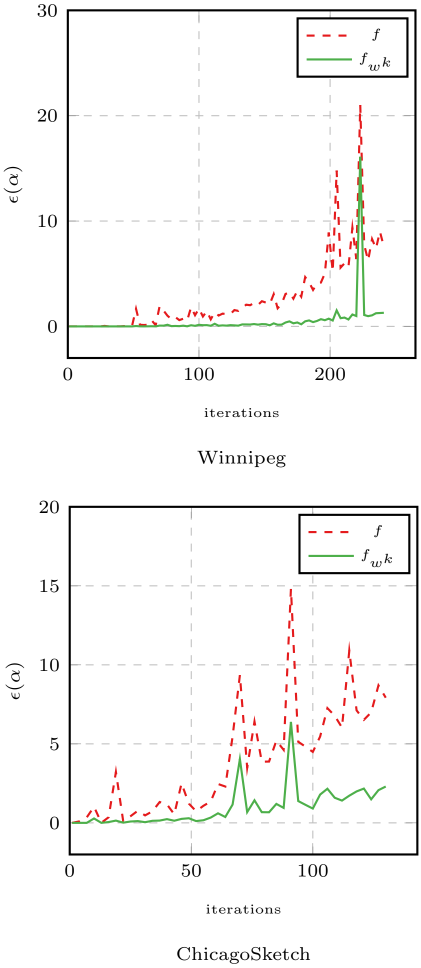
value and the error $\epsilon(f)$ of an evaluation is computed as follows

$$\epsilon(f) = \frac{|f^{\infty} - f|}{f^{\infty}}.$$

In Figure 4.2 we can observe that using f_{w^k} leads to more accurate evaluations than that of f. As a consequence, with a robust evaluation of the monotone line search rule accomplished by the use of f_{w^k} described in Section 4.4.1, the precision of the step α^k is greater than that of using f, as it is reported in Figure 4.3, where $\epsilon(\alpha)$ is computed as follows

$$\epsilon(\alpha) = \frac{|\alpha^\infty - \alpha|}{\alpha^\infty}$$

and α^∞ is obtained using f^∞ in the line search procedure. Note that in



Winnipeg

ChicagoSketch

Figure 4.3: Linesearch step computation error with f and f_{w^k} in the line search procedure through iterations.

Figure 4.3 a few error spikes can be observed even when employing f_{w^k}, in correspondence with the higher ones obtained by employing f. This behavior could be a consequence of the use of the quadruple precision as the ground truth of the step length value. In fact, it may perform a few more reductions of α due to the large amount of significant digits provided by the representation.

In the nonmonotone case, the line search step precision is affected by the precision of Δ^k_{max} defined in (4.43). In Figure 4.4, it is reported the error

$$\epsilon(\Delta_{max}) = \frac{|\Delta^{\infty}_{max} - \Delta_{max}|}{\Delta^{\infty}_{max}}$$

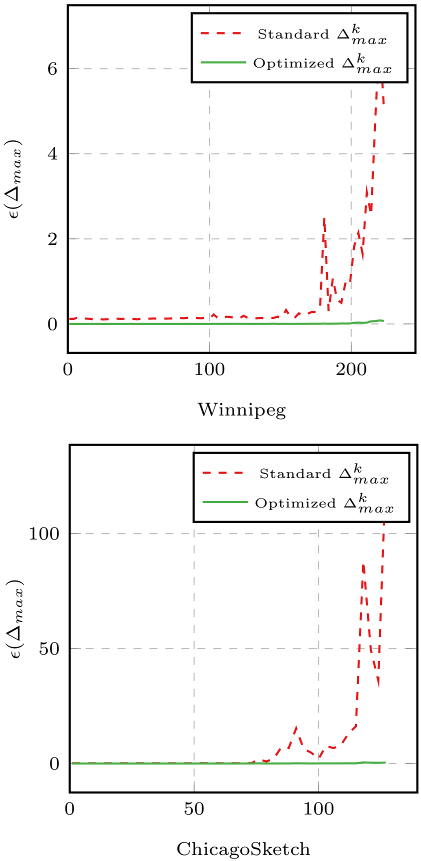
obtained using the standard $\Delta_{max} = f^k_{max} - f(x^k)$ and the optimized one from (4.44), where Δ^{∞}_{max} is obtained by means of quadruple precision evaluations. From Figure 4.4, we can observe that using the standard Δ_{max} leads quickly to noisy values affected by the cancellation error that occurs when computing the difference between two closed values in terms of significant digits. With the optimized Δ_{max} given in (4.44), the obtained precision is high even in later iterations, allowing the nonmonotone strategy to be employed in a robust manner.

As a consequence, the error of the nonmonotone step α^k computation behaves accordingly with the Δ_{max} precision, as it is widely confirmed in Figure 4.5. Using $f_w(x)$ in a separable function like the one considered in this section leads to relevant benefits both in terms of computational time and numerical precision, in fact allowing the exploiting of the nonmonotone strategy even in large-scale problems.

Note that the effect of the numerical tricks described in Section 4.4 is greater as the network size increases. That's why we were able to perform experiments on large networks such as Philadelphia, ChicagoRegional and Sydney where even monotone methods have to face with numerical issues in later stages of the optimization. However, we were not able to show the numerical advantages on these networks due to the high computational effort required by the quadruple precision floating-point representation, which is software simulated.

4.6 Conclusions

In this work, a general nonmonotone decomposition framework was defined for smooth constrained optimization problems. The need of adopting non-

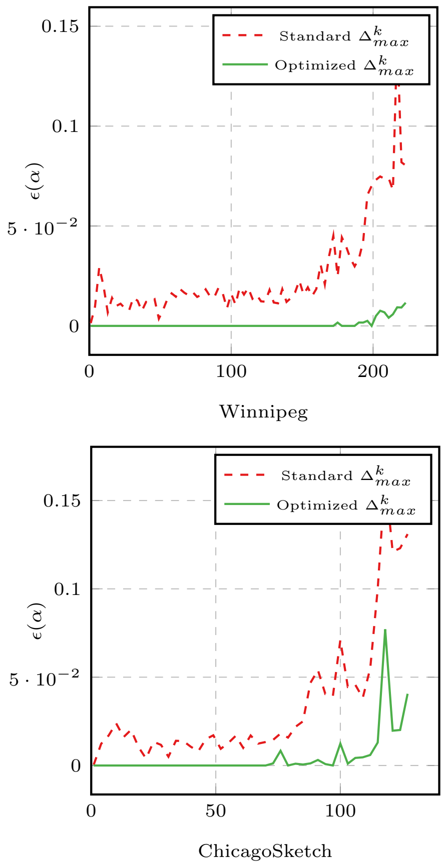


Winnipeg

ChicagoSketch

Figure 4.4: Standard and optimized Δ^k_{max} computation error in the nonmonotone line search procedure through iterations.

monotone strategies within a decomposition framework is motivated by the fact that block of variables are individually updated, so that, it could be computationally advantageous to avoid of enforcing a monotone reduction at each iteration. The main contributions of the chapter are: 1) the development of a unified convergence theory that, with respect to [16], includes the possibility of adopting nonmonotone line search mappings; 2) the definition of a nonmonotone Gauss-Soutwell decomposition algorithm for classes of problems with a single equality constraint and box constraints that arise in several contexts; 3) the definition of a nonmonotone decomposition technique for facing the numerical issues arising in the context of large-scale



Winnipeg

ChicagoSketch

Figure 4.5: Line search step computation error with standard and optimized Δ^k_{max} in the nonmonotone line search procedure through iterations.

problems with partially separable objective functions. The numerical experiments performed on large-scale network equilibrium problems have shown the efficiency of the nonmonotone strategy with respect to the standard monotone counterpart, and the importance of employing the proposed nonmonotone techniques for dealing with numerical errors.

4.7 Appendix: Proof of Lemma 4

Proof. First, we show that the sequence $\{f^k_{max}\}$ is monotonically nonincreasing. Let us recall here that condition (4.9) might be written as

$$f(x^{k+1}) \le f(x^{r(k)}) - \sigma(\|x^{k+1} - x^k\|), \tag{4.46}$$

where $r(k)$ is defined in (4.8). Since $\min(k+1, W) \le \min(k, W) + 1$, we have

$$\begin{aligned} f(x^{r(k+1)}) &= \max_{0\le j\le \min(k+1,W)} f(x^{k-j+1}) \\ &\le \max_{0\le j\le \min(k,W)+1} f(x^{k-j+1}) \\ &= \max\{f(x^{r(k)}), f(x^{k+1})\} = f(x^{r(k)}), \end{aligned}$$

where last equality follows from (4.46). Since $\{f(x^{r(k)})\}$ is nonincreasing and $x^{r(0)} = x^0$, we have that $f(x^k) \le f(x^0)\ \forall k$, which proves (a).

Since f is limited from below, the monotone nonincreasing sequence $\{f(x^{r(k)})\}$ admits a limit W^* for $k \to \infty$. By induction on j, with $1 \le j \le W+1$, let us prove that the two limits below are satisfied:

$$\lim_{k\to\infty} \|x^{r(k)-j+1} - x^{r(k)-j}\| = 0 \tag{4.47}$$

$$\lim_{k\to\infty} f(x^{r(k)-j}) = \lim_{k\to\infty} f(x^{r(k)}) \tag{4.48}$$

where k is assumed to be large enough to have $r(k) \ge k - W > 1$.

If $j = 1$, using (4.46) with $k = r(k) - 1$, we have

$$f(x^{r(k)}) \le f(x^{r(r(k)-1)}) - \sigma(\|x^{r(k)} - x^{r(k)-1}\|).$$

Thus, together with convergence of $\{f(x^{r(k)})\}$, we obtain

$$\lim_{k\to\infty} \|x^{r(k)} - x^{r(k)-1}\| = 0$$

From Lipschitz continuity of f and the above limit we obtain that

$$\lim_{k\to\infty} f(x^{r(k)-1}) = \lim_{k\to\infty} f(x^{r(k)}),$$

which means that induction has been proved for the case $j = 1$.

Now assume that (4.47) and (4.48) are valid for a given j. From (4.46) used with
$k = r(k) - j - 1$, we have that

$$f(x^{r(k)-j}) \le f(x^{r(r(k)-j-1)}) - \sigma(\|x^{r(k)-j} - x^{r(k)-j-1}\|).$$

Thus, together with (4.48), we obtain that

$$\lim_{k\to\infty} \|x^{r(k)-j} - x^{r(k)-j-1}\| = 0.$$

From the limit above, Lipschitz continuity of f and again (4.48) we obtain that

$$\lim_{k\to\infty} f(x^{r(k)-j-1}) = \lim_{k\to\infty} f(x^{r(k)}),$$

which means that induction has been proved from a generic j to $j+1$.

In particular (4.47) and (4.48) are also valid if we replace $r(k)$ with $R(k) := r(k+W+1)$. Now, for k sufficiently large, we have that

$$\begin{aligned} x^{R(k)} &= x^k + (x^{k+1} - x^k) + \cdots + (x^{R(k)} - x^{R(k)-1}) \\ &= x^k + \sum_{j=1}^{R(k)-k} \left(x^{R(k)-j+1} - x^{R(k)-j}\right) \end{aligned} \tag{4.49}$$

Since $r(k+W+1) \leq k+W+1$, we have $R(k) - k \leq W+1$ and, thus, from (4.49) and (4.47) used replacing $r(k)$ with $R(k)$, we obtain

$$\lim_{k\to\infty} \|x^k - x^{R(k)}\| = 0.$$

From convergence of $\{f(x^{r(k)})\}$ and Lipschitz continuity, it follows that

$$\lim_{k\to\infty} f(x^k) = \lim_{k\to\infty} f(x^{R(k)}) = \lim_{k\to\infty} f(x^{r(k+W+1)}) = W^*,$$

which complete proof of (b). Thesis (c) follows from (b) and (4.46). □

4.8 Appendix: Proof of Proposition 3

Proof. By contradiction assume that there exists an infinite subset K such that for all $k \in K$ we have

$$\beta^k \nabla f(x^k)^T d^k \leq -\nu < 0. \tag{4.50}$$

Assume that there exists an infinite subsequence $K_1 \subseteq K$ such that $\forall\ k\ \in K_1$ we have that $\alpha^k = \beta^k$. From (a) and (c) of Lemma 4 we have that

$$\lim_{k\to\infty, k\in K_1} \beta^k \nabla f(x^k)^T d^k = 0,$$

which contradicts (4.50).
Assume now that there exists a $\hat{k}$ such that $\forall\ k\ \in K, k > \hat{k}$ we have that $\alpha^k < \beta^k$. Thus, from the line search instructions, we have that

$$f(x^k + \frac{\alpha^k}{\delta} d^k) > f^k_{max} + \gamma \frac{\alpha^k}{\delta} \nabla f(x^k)^T d^k \geq f(x^k) + \gamma \frac{\alpha^k}{\delta} \nabla f(x^k)^T d^k.$$

From the Mean Value Theorem, applied on the above inequality, we obtain that

$$-\gamma \nabla f(x^k)^T d^k > -\nabla f(z^k)^T d^k, \tag{4.51}$$

where $z^k = x^k + \theta^k \frac{\alpha^k}{\delta} d^k$, with $\theta^k \in [0,1]$. From Lemma 4 we get that if $\{x^k\}$ converges to $\bar{x}$, also $\{z^k\}$ converges to $\bar{x}$. Since $\{d^k\}$ is bounded (by Assumption 8), we get that there exists a subsequence, redefined $\{d^k\}$, that converges to $\bar{d}$. Then, taking the limits for $k \in K$ and $k \to \infty$, we obtain

$$(1-\gamma)\nabla f(\bar{x})^T \bar{d} \geq 0,$$

and this contradicts (4.50) being $\beta^k > 0$.

□

4.9 Appendix: Proof of Proposition 4

Proof. By contradiction assume that there exists an infinite subset K such that for all $k \in K$ we have

$$\beta^k \nabla f(x^k)^T d^k \leq -\nu < 0. \tag{4.52}$$

Assume that there exists an infinite subsequence $K_1 \subseteq K$ such that $\forall\ k\ \in K_1$ we have that $\alpha^k = \beta^k$. From (a) and (c) of Lemma 4 we have that

$$\lim_{k\to\infty, k\in K_1} \beta^k \nabla f(x^k)^T d^k = 0,$$

which contradicts (4.52).
Assume now that there exists a $\hat{k}$ such that $\forall\ k\ \in K, k > \hat{k}$ we have that $\alpha^k < \beta^k$. Thus, from the line search instructions, we have that

$$f(x^k + \frac{\alpha^k}{\delta} d^k) > f^k_{max} - \gamma \| \frac{\alpha^k}{\delta} d^k \|^2 \geq f(x^k) - \gamma \| \frac{\alpha^k}{\delta} d^k \|^2.$$

From the Mean Value Theorem, applied on the above inequality, we obtain that

$$\frac{\gamma}{\delta} \alpha^k \|d^k\|^2 > -\nabla f(z^k)^T d^k \tag{4.53}$$

where $z^k = x^k + \theta^k \frac{\alpha^k}{\delta} d^k$, with $\theta^k \in [0,1]$. From Lemma 4 we get that if $\{x^k\}$ converges to $\bar{x}$, also $\{z^k\}$ converges to $\bar{x}$. Since $\{d^k\}$ is bounded (by Assumption 9), we get that there exists a subsequence, redefined $\{d^k\}$, that converges to $\bar{d}$. Then, taking the limits for $k \in K$ and $k \to \infty$, we obtain

$$\nabla f(\bar{x})^T \bar{d} \geq 0,$$

and this contradicts (4.52) being $\beta^k > 0$. □

4.10 Appendix: Proof of Proposition 5

Proof. From the properties of the projection mapping we get that

$$\|d^k\| = \|P_{\mathcal{F}_{h^k}}[x^k_{(h^k)} - \nabla_{(h^k)} f(x^k)] - x^k_{(h^k)}\| \leq \|\nabla_{(h^k)} f(x^k)\| \leq \|\nabla f(x^k)\|$$

and

$$\nabla f(x^k)^T d^k \leq -\|P_{\mathcal{F}_{h^k}}[x^k_{(h^k)} - \nabla_{(h^k)} f(x^k)] - x^k_{(h^k)}\|^2 = -\|d^k\|^2. \quad (4.54)$$

These inequalities, together with compactness of $\mathcal{F}$ prove that Assumptions 8 and 9 are satisfied. Let $\{x^k\}$ be a convergent sequence, from Proposition 3 (or 4) and the fact that $\beta^k \geq 1$, we have that Assumption 7 is satisfied.

Now let us prove Assumption 6. Let $\{x^k\}$ be a subsequence convergent to x^*. From the continuity of the projection operator, Assumption 7 and (4.54) we get that

$$\lim_{k\to\infty} -\nabla f(x^k)^T d^k \geq \|P_{\mathcal{F}_h}[x^*_{(h)} - \nabla_{(h)} f(x^*)] - x^*_{(h)}\| = 0,$$

which implies that

$$x^*_{(h)} = P_{\mathcal{F}_h}[x^*_{(h)} - \nabla_{(h)} f(x^*)]$$

and, in turn, that

$$\nabla_{(h)} f(x^*)^T \left(x_{(h)} - x^*_{(h)}\right) \geq 0 \quad \forall\, x_{(h)} \in \mathcal{F}_h.$$

□

4.11 Appendix: Proof of Proposition 6

Proof. Since $\mathcal{F}$ is compact, $\|d^k\| = \|\hat{x}^k_{(h^k)} - x^k_{(h^k)}\|$ is always bounded. From (4.15) we also have that

$$\nabla f(x^k)^T d^k = \nabla_{(h^k)} f(x^k)^T \left(\hat{x}^k_{(h^k)} - x^k_{(h^k)}\right) < 0,$$

otherwise a null step will be instead considered. This proves that Assumption 9 is satisfied.

Let $\{x^k\}$ be a convergent sequence, from Proposition 4 and the fact that $\beta^k \geq 1$ we get that Assumption 7 is satisfied.

Now let us prove Assumption 6. Let $\{x^k\}$ be a subsequence convergent to x^*. The direction d^k is such that, for every $x_{(h)} \in \mathcal{F}_h$

$$\nabla_{(h)} f(x^k)^T d^k \leq \nabla_{(h)} f(x^k)^T \left(x_{(h)} - x^k_{(h)} \right).$$

Thus, from the continuity of the gradient and Assumption 7, we get that

$$\nabla_{(h)} f(x^*)^T \left(x_{(h)} - x^*_{(h)} \right) \geq 0, \quad \forall x_{(h)} \in \mathcal{F}_h.$$

□

Chapter 5

Controlling nonmonotonicity: a new line search family

In this chapter we design a new family of nonmonotone line searches that has direct control over the degree of nonmonotonicity introduced. This new idea comes from the identification of a significant connection between the nonmonotone technique by Grippo, Lampariello and Lucidi [55] and the more recent one by Li and Fukushima [70]. The proposed framework combines the two strategies to obtain the best from both of them: a technique that has direct control over the amount of nonmonotonicity, but which is still easy to be designed. The numerical analysis conducted on a set of large-scale network equilibrium problems shows an example in which controlling the amount of nonmonotonicity is very helpful for both improving results and obtaining a better understanding on the preferred paths of the optimization procedure.[1]

[1] This chapter is based on a paper which is now under review "Controlling the degree of nonmonotonicity: a new line search framework combining two nonmonotone techniques" in *Operational Research Letters.*

5.1 Introduction

Let us consider the problem

$$\begin{aligned} &\min_x f(x), \\ &\text{s.t. } x \in \mathbb{R}^n, \end{aligned} \tag{5.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. We focus on solving (5.1) with a line search based optimization method. Without loss of generality we assume to employ a Newton-like method globalized by an Armijo line search (see Algorithm 5). Note that the choices of the direction and of the line search are not crucial for the aim of this work, since the focus here is on nonmonotone techniques. For sake of simplicity we'll focus on unconstrained optimization, but extending the discussion to the constrained case is very easy.

```
Algorithm 5: Line search based Algorithm
Input: $x^0 \in \mathbb{R}^n$ starting point
$k = 0$
while stopping criterion is not fulfilled do
    define a descent direction $d^k$ (e.g. a Newton direction)
    compute a step length $\alpha^k$ by means of a suitable line search (e.g.
      Armijo line search) along $d^k$,
    $x^{k+1} = x^k + \alpha^k d^k$
    $k = k + 1$
```

The first nonmonotone technique was proposed by Grippo, Lampariello, Lucidi [55] to globalize the Newton method without enforcing monotonicity on the resulting $\{f(x^k)\}$ sequence. In fact, the monotone requirement might slow the speed of convergence of many fast local optimization methods, spectral gradient (or Barzilai-Borwein) in [82], spectral projected gradient in [11], sequential quadratic programming in [105], Polak-Ribière-Polyak in [106]. This is the reason why nonmonotone techniques are very well suited for all those combinations of directions and initial step sizes that are showed to work really well in practice, e.g. Newton direction and unit step size, anti-gradient and spectral step size, etc. In concrete, nonmonotone strategies help avoid the reduction of the step size caused by the monotone requirement in backtracking line searches. Nonmonotone line searches are in fact designed to accept the initial step size more often than their monotone counter-part. Thus, the *nonmonotonicity* is the tool given to algorithm developers to influence how often accept an initial step size.

In the field of nonmonotone techniques, it is still not clear in which phase of the optimization process *nonmonotonicity* should be empathized or reduced. In [55] they claim that *nonmonotonicity* is particularly valuable in intermediate and final stages. This is not in contrast with [90, 91] in which *nonmonotonicity* is not applied in the first iterations. On the other side, in [2, 100] they claim that in the beginning there is need for more *nonmonotonicity* while the opposite happens when the algorithm approaches a solution. This is also supported for instance by [57, 78]. Another different point of view is that of [20, 54, 58, 67], in which *nonmonotonicity* is needed both in the beginning and in the end of the optimization procedure.

The above different conclusions suggest that the amount of *nonmonotonicity* required is highly dependent on the application and on the optimization method. On the other hand, it is also clear that a lot more can be done in this field, since nonmonotone techniques have often been applied just to improve existing results, but not much has been done in trying to understand in which applications/methods/phases *nonmonotonicity* is more needed. This systematic approach is probably missing because of the lack of three elements: a review on nonmonotone techniques, a common set of test problems on which to compare them and an easy interpretable tool that might be exploited to point out common behaviors. In this work we will focus on the third of these elements: providing a new family of nonmonotone techniques that has direct control over the degree of *nonmonotonicity* w.r.t. the optimization phase. This idea originates from the identification of a significant connection between two very popular nonmonotone techniques: the original one from [55] and the more recent one from [70].

To the best of our knowledge, the theme of controlling the degree of nonmonotonicity is central only in another recent work [79]. In particular they propose two adaptive nonmonotone rules based on the morphology of the objective function. One of the techniques adaptively updates W almost as in [80], while the other one computes a convex combination of function value, as in [100], but taking into account only a subsequence of $\{f(x^k)\}$. As a matter of fact, the focus of [79] is mainly on obtaining a more efficient algorithm, rather than studying which is degree of nonmonotonicity that is more appropriate in each phase of the optimization process.

In the next section we will first show the connection between [55] and [70], then we will derive the new family of nonmonotone strategies. In Section 5.3 the global convergence result is discussed. An illustrative example of the application of such a framework to a set of network equilibrium problems is presented in Section 5.4. Conclusions are drawn in Section 5.5.

Regarding notation, with $\|.\|$ we indicate the Euclidean norm, $\mathcal{L}_0 = \{x \in \mathbb{R}^n : f(x) \leq f(x^0)\}$ is the 0-level set, $\mathbb{R}_+$ is the positive orthant, $\sigma : \mathbb{R}_+ \to \mathbb{R}_+$ is a forcing function, i.e., for each sequence $\{t^k\}$ with $t^k \in \mathbb{R}_+$ we have that

$$\lim_{k\to\infty} \sigma(t^k) = 0 \quad \Rightarrow \quad \lim_{k\to\infty} t^k = 0.$$

5.2 A new family of nonmonotone techniques

Given $\delta \in (0,1)$ and α_0^k the initial step size at iteration k, the monotone Armijo line search finds the largest $\alpha^k \in \{\alpha_0^k, \alpha_0^k\delta, \alpha_0^k\delta^2, \dots\}$ such that the following monotone condition is satisfied

$$f(x^k + \alpha^k d^k) \leq f(x^k) + \gamma\alpha^k \nabla f(x^k)^T d^k, \tag{5.2}$$

where $\gamma \in (0,1)$ and d^k is the direction computed at iteration k. Now, let $W > 0$ be the nonmonotone window, then we define

$$f_{max}^k := \max_{k-W\leq i\leq k} f(x^i) \tag{5.3}$$

as the largest function value among the last few iterations, where, formally, we set $x^{-1} := x^{-2} := \dots := x^{-W} := x^0$ (or, alternatively, we can replace W by $W^k := \min\{k, W\}$). Furthermore, let us define

$$r(k) \leq k \text{ as the (largest) iteration index such that } f(x^{r(k)}) = f_{max} \tag{5.4}$$

holds. The aim of a nonmonotone technique applied on Armijo line search is that of relaxing condition (5.2). In particular in [55], it was proposed to replace $f(x^k)$ with f_{max}^k and obtain

$$f(x^k + \alpha^k d^k) \leq f_{max}^k + \gamma\alpha^k \nabla f(x^k)^T d^k. \tag{5.5}$$

The degree of *nonmonotonicity* in (5.5) is controlled by the parameter W. Numerical results obtained by employing (5.5) on some specific problems might be highly influenced by the choice of W ([55, 82, 90]). But W is a pretty indirect way to control *nonmonotonicity*. For instance whenever a high amount of *nonmonotonicity* is needed in the early stage of the optimization procedure, modifying W would not be of any help: by definition, no $f(x^{r(k)})$ can be greater than $f(x^0)$. In fact, (5.5) and the majority of nonmonotone techniques ([20,80,90,97,100]) enforces $\{x^k\}$ to remain in $\mathcal{L}_0$, while the starting point x^0 might be an arbitrary bad guess from which to start. On the other side in the late phase of the optimization procedure, f^k_{max} might be too big or too close to $f(x^k)$. This issue is also very difficult to be solved by only controlling W, especially if a single W needs to be chosen for a whole set of problems.

To solve the limitation of keeping $\{x^k\}$ within $\mathcal{L}_0$, in [54] and [58] are developed algorithms that give $\{x^k\}$ a lot more freedom. In particular, (5.5) (or similar) is not checked at each iteration, but only every $L > 0$ iterations. Let us call z^j all those points on which condition (5.5) is not checked: the algorithm exploits z^j to explore regions which do not belong to $\mathcal{L}_0$. In [54], to ensure strong global convergence properties, a backtracking procedure is needed to come back to a “safe” iterate x^k. Thanks to the change of notation (exploration points z^j are not saved in the sequence $\{x^k\}$), the algorithm proposed in [54] is also able to keep $\{x^k\}$ within $\mathcal{L}_0$.

Another nonmonotone technique that is numerically very interesting was originally proposed in [70] and the modified condition is the following

$$f(x^k + \alpha^k d^k) \leq f(x^k) + \epsilon^k + \gamma \alpha^k \nabla f(x^k)^T d^k, \tag{5.6}$$

where $\{\epsilon^k\}$ is a positive sequence satisfying

$$\sum_{k=0}^{\infty} \epsilon^k < \epsilon < \infty. \tag{5.7}$$

If we employ (5.6), it is still possible to obtain strong global convergence results and $\{x^k\}$ might be proved to belong to $\mathcal{L}_{0+\epsilon} = \{x \in \mathbb{R}^n : f(x) \leq f(x^0) + \epsilon\}$. In this case, the ϵ-sequence is defined by the user and the only requirement is (5.7). This means that there is an high control over the degree of *nonmonotonicity* that can be introduced in each phase. This great amount of freedom is also the biggest drawback of (5.6). In fact, it is very hard to design an ϵ-sequence that has the right amount of *nonmonotonicity* in each phase. This is totally unpractical when a single sequence has to be designed for a whole set of problems.

A different point of view on (5.5) is now discussed. The idea is looking at the sequence $\{f^k_{max} - f(x^k)\}$ as a particular $\{\epsilon^k\}$ sequence. Let us define

$\Delta^k := f^k_{max} - f(x^k)$ and write again (5.5) using Δ^k:

$$f(x^k + \alpha^k d^k) \leq f(x^k) + \Delta^k + \gamma \alpha^k \nabla f(x^k)^T d^k. \tag{5.8}$$

Note that $\Delta^k \geq 0$, by definition. Condition (5.8) looks exactly like (5.6), except from the fact that the sequence $\{\Delta^k\}$ is automatically derived from the algorithm and it self-adapts to each run, while $\{\epsilon^k\}$ is a sequence defined by the user. This connection between [55] and [70] can be exploited to define a new family of nonmonotone techniques that combines both of them.

To have direct control over the degree of *nonmonotonicity*, a modification on (5.8) is now proposed. Let us define the function $g : \mathbb{R}_+ \to \mathbb{R}_+$ as follows

$$g(t) = \begin{cases} t^{\beta} & t > 1, \\ t^{\frac{1}{\beta}} & t \leq 1, \end{cases} \tag{5.9}$$

where $\beta > 0$. The new condition can be defined as follows

$$f(x^k + \alpha^k d^k) \leq f(x^k) + g(\Delta^k) + \gamma \alpha^k \nabla f(x^k)^T d^k. \tag{5.10}$$

Thanks to (5.10) is now possible to control this, for instance if the application requires an high degree of *nonmonotonicity* we can for example use $\beta = 2$. This for instance would give the algorithm enough freedom to avoid the limitation of keeping $\{x^k\}$ within $\mathcal{L}_0$, requirement that is not relaxed in [79, 80, 100]. On the other hand, if the nonmonotone effect required is low we can for example use $\beta = 0.5$. The constant β has in fact a direct effect on the amount of *nonmonotonicity* we intend to employ. Thanks to this idea we have great freedom for designing the controlling sequence $\{g(\Delta^k)\}$, but at the same time the nonmonotone effect now directly depends on $f(x^k)$. In Section 5.4 we will discuss how to use (5.10) to both improve performances of the nonmonotone algorithm and extract interesting insights on the amount of *nonmonotonicity* needed in the different phases.

Note that if we directly employ g with $\beta = 2$, global convergence is not guaranteed. In the following section we will focus on some theoretical aspects related to (5.10), starting from original results obtained for (5.5) and (5.6).

5.3 Global Convergence

To prove global convergence of Algorithm 5 in which (5.5) or (5.6) is employed on Step 4, we need the following assumption on the direction.

Assumption 10. Let $\{d^k\}$ be a sequence of feasible search directions. Assume that

(a) $\forall\, k$ we have $\nabla f(x^k)^T d^k \leq -c_1 \|\nabla f(x^k)\|^2$ for a given constant $c_1 > 0$,

(b) $\forall\, k$ we have $\|d^k\| \leq c_2 \|\nabla f(x^k)\|$ for a given constant $c_2 > 0$.

Two key results to obtain convergence using (5.5) or (5.6) are respectively Lemma 8 and Lemma 9.

Lemma 8. Suppose that $\mathcal{L}_0$ is compact. Let $\{x^k\}$ be a sequence of points such that

$$f(x^{k+1}) \leq M^k - \sigma(\|x^{k+1} - x^k\|), \tag{5.11}$$

where $f(x^k) \leq M^k \leq f^k_{max}$ and σ is a forcing function, then we have

(a) $x^k \in \mathcal{L}_0$;

(b) $\{f^k_{max}\}$ and $\{f^k\}$ converge to the same limit;

(c) $\lim\limits_{k\to\infty} \|x^{k+1} - x^k\| = \lim\limits_{k\to\infty} \|\alpha^k d^k\| = 0$.

See Lemma 1 of [54] for the proof. Note that (5.11) is more general than (5.5), not only because $\nabla f(x^k)^T d^k$ is replaced by $\sigma(\|x^{k+1} - x^k\|)$, but in particular because M^k is any value between $f(x^k)$ and f^k_{max}.

Lemma 9. Suppose that $\mathcal{L}_{0+\epsilon}$ is compact. Let $\{x^k\}$ be a sequence of points such that

$$f(x^{k+1}) \leq f(x^k) + \epsilon^k - \sigma(\|x^{k+1} - x^k\|), \tag{5.12}$$

where $\sum_{k=0}^{\infty} \epsilon^k < \epsilon < \infty$. and σ is a forcing function, then we have

(a) $x^k \in \mathcal{L}_{0+\epsilon}$;

(b) $\lim\limits_{k\to\infty} \|x^{k+1} - x^k\| = \lim\limits_{k\to\infty} \|\alpha^k d^k\| = 0$.

In most of the nonmonotone settings one key result is $\lim\limits_{k\to\infty} \|x^{k+1} - x^k\| = 0$. The main difference between Lemma 8 and Lemma 9 lies in the way this result is obtained: using (5.5) we first need to prove that $\{f^k_{max}\}$ and $\{f^k\}$ converge to the same limit, while using (5.6) the result directly follows from (5.7). To prove Lemma 8 it is very important to have $f(x^{r(k)}) \geq f(x^{k+1})$, while we do not have a similar result for the case of (5.6). On the other side, (5.7) and the fact that $\{\epsilon^k\}$ is user-defined make Lemma 9 almost a trivial result. The next result is the main convergence theorem.

Theorem 3. Let $\{d^k\}$ be a sequence of search directions satisfying Assumption 10 and assume that $\mathcal{L}_{0+\epsilon}$ is compact. Let $\{x^k\}$ be a sequence of points generated by Algorithm 5, in which line search condition of Step 4 is either (5.11) or (5.12). Then, if $\{x^k\}$ converges to $\bar{x}$, we have

$$\lim_{k\to\infty} \nabla f(x^k) = 0. \tag{5.13}$$

Now that existing results have been recalled for both conditions (5.5) and (5.6), we can discuss how to obtain global convergence also for (5.10). From definition of W^k in (5.11) we directly have convergence also for (5.10) whenever (5.9) is employed with $\beta \leq 1$. In addition, the same convergence properties are directly achieved for an arbitrary function $g : \mathbb{R}_+ \to \mathbb{R}_+$, whenever $g(t) \leq t$. On the other hand, if we employ a function g in which $g(t) > t$ (i.e., g defined in (5.9) with $\beta > 1$), to obtain convergence we need some additional assumptions (see for instance [67]).

To avoid any additional assumptions and still get the same result, a simpler way is instead that of embedding (5.10) into (5.6). Consider for instance the sequence

$$\epsilon^k = \min\{g(\Delta^k), \frac{B}{\omega^k}\}, \tag{5.14}$$

where $B > 0$ is a large constant, $\omega > 1$ is a constant close to 1 and g is defined by (5.9) with $\beta = 2$. Thanks to the fact that there exists an $\epsilon > 0$ such that

$$\sum_{k=0}^{\infty} \frac{B}{\omega^k} = \frac{B}{1-\omega} \leq \epsilon < \infty,$$

we have that (5.7) is satisfied. Thus, global convergence is proved for Algorithm 5 if in Step 4 it is employed the condition

$$f(x^{k+1}) \leq f(x^k) + \min\{g(\Delta^k), \frac{W}{\omega^k}\} - \sigma(\|x^{k+1} - x^k\|), \tag{5.15}$$

where $g : \mathbb{R}_+ \to \mathbb{R}_+$ is an arbitrary function.

Remark 3. Note that condition (5.15) is ensuring that $\{x^k\}$ is kept within $\mathcal{L}_{0+\epsilon}$ and not within $\mathcal{L}_0$. The same might be said for the original condition (5.6), which was showed to achieve good numerical performances when the $\{\epsilon^k\}$ sequence is well designed. In addition, if the objective function f is convex, it can easily be proved that both (5.6) and (5.15) lead limit points of $\{x^k\}$ to belong to $\mathcal{L}_0$.

Remark 4. As anticipated, Theorem 3 is obtained in the case of unconstrained optimization. The result might be extended to the constrained case as follows:

- the initial step size α_0^k is assumed to be feasible;
- defined β^k the maximum feasible step size along the direction d^k, from Theorem 3 we now get something weaker than (5.13): $\lim_{k\to\infty} \beta^k \nabla f(x^k)^T d^k = 0$;
- to obtain (5.13) we need to exploit some addition properties on the direction (e.g., if the direction is always feasible we get that β^k is limited from below $\Rightarrow$ (5.13)).

5.4 Illustrative example

In this section we report an illustrative example of the use of (5.15) on a class of convex problems: Network Equilibrium (NE) problems. This class has attracted the attentions of many researchers in the optimization field because they are nonacademic large-scale problems in which the accuracy of the solutions is often very important. Note that the results below are not presented to show the supremacy of the proposed condition (5.15) in a general setting. The aim is rather that of showing an example in which having great freedom in designing (5.15) is helpful not only to improve results, but also to extract interesting insights on the application.

Let $G = (N, A)$ be a graph, where N is the set of nodes and A is the set of arcs. Let $P \subseteq N \times N$ be the set of all Origin/Destination (O/D) pairs ($L := |P|$). For each $h \in P$, a travel demand $D_h > 0$ is defined. Given such a network, the NE problem concerns the forecasting of the flow on each arc. We take into account the path-based formulation of the NE (see [29] for a complete description). In particular, indicated by x the vector of path flows, the objective function is a smooth, convex function defined by

$$f(x) = \sum_{a\in A} \int_0^{v_a(x)} s_a(t)\, dt,$$

where $v_a(x)$ is the total flow on arc a and $s_a(\cdot)$ is the arc-separable cost. The resulting optimization problem is the following

$$\begin{aligned}&\min f(x),\\&\text{s.t. } x \in \mathcal{F}_1 \times \cdots \times \mathcal{F}_L,\end{aligned}$$

where, given the O/D pair h, $\mathcal{F}_h = \{x_{(h)} \in \mathbb{R}^{n_h} : \sum_{i\in\{1,\ldots,n_h\}} x_{(h),i} = D_h,\ x_{(h)} \geq 0\}$, n_h is the total number of paths insisting on h and $x_{(h),i}$ is the flow on the i-th path of h. Note that the above formulation might be considered "virtual", since the total number of paths is exponential in the dimension of the network. For this reason it is usually defined a restricted set in which only paths with nonzero flow are considered. Then, variables are iteratively added by means of a column generation technique.

To solve the NE problem, Algorithm 5 is modified by considering a decomposing method along each O/D pair and including the above discussed column generation technique. Step 3 is defined considering only 2 nonzero components: respectively the path with the highest and lowest cost. Flow will be removed from the first path and added to the second. In Step 4 we employ condition (5.15), where various implementations for g are considered and defined below.

As regards the computation of the initial step length along the search direction, a commonly used choice [18] is the one obtained by minimizing the quadratic approximation of f along the direction. Taking into account that the direction has only 2 nonzero components and that the partial derivatives of f represent the costs of the corresponding paths, the closed-form solution of the above step length can be easily and efficiently calculated. This step length was showed to perform well in practice if employed directly, but to ensure global convergence a line search mapping is needed. In the monotone setting, it might happen that a sufficient decrement is not obtained employing this step, resulting in at least one reduction of it. This is the precise context for which nonmonotone techniques were designed: accepting as often as possible a good initial step without reducing it.

The test problems used for the experiments are freely available at the web page `http://www.bgu.ac.il/~bargera/tntp/` and are described in Table 5.1. All the algorithms have been implemented in `C++` and parameters chosen for the line searches are

$$\delta = 0.5, \quad \gamma = 10^{-2}, \quad W = 10.$$

Since numerically we have that $k < \infty$, we can chose W to be very large and ω to be very close to 1. Thus, in practice we can obtain

$$\epsilon^k = \min\{g(\Delta^k), \frac{B}{\omega^k}\} = g(\Delta^k),$$

by setting for example $\omega = 1 + 10^{-8}$ and $B = 10^{10}$.

Network	Label	# links	# nodes	# centroids	# O/D pairs
Sioux-Falls	SF	76	24	24	528
Winnipeg	W	2,535	1,067	154	4,345
Barcelona	B	2,522	1,020	110	7,922
Chicago-Sketch	CS	2,950	933	387	93,135
Berlin-Center	BC	28,376	12,981	865	49,688
Philadelphia	P	40,003	13,389	1,525	1,149,795
Chicago-Regional	CR	39,018	12,982	1,790	2,296,227
Sydney	S	75,379	33,113	3,264	3,340,619

Table 5.1: Network datasets details

In this context we compare Algorithm 5 (slightly modified as described above) where in Step 4 the implementations of g are the following:

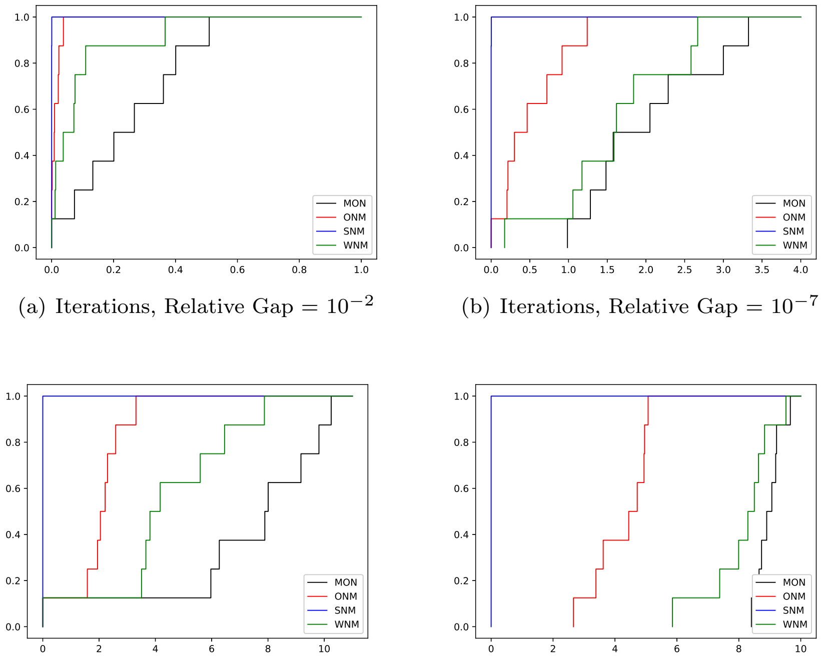
- MONotone (MON): $g(t) = 0$,
- Original NonMonotone (ONM): $g(t) = t$,
- Strongly NonMonotone (SNM): $g(t)$ as in (5.9), with $\beta = 2$,
- Weakly NonMonotone (WNM): $g(t)$ as in (5.9), with $\beta = \frac{1}{2}$.

Note that SNM is always introducing more *nonmonotonicity* than ONM, while WNM is always reducing it. Now, to understand in which phases and how often we have

$$\Delta^k = f^k_{max} - f(x^k) \leq 1, \tag{5.16}$$

executions of ONM have been checked:

- in the initial phase of the optimization process (first 1000 iterations), (5.16) is never satisfied in more than 7% of the iterations (in most of the executions less than 1%);
- in the late phase of the optimization process (last 1000 iterations), (5.16) is always satisfied in the 100% of the iterations;



(a) Iterations, Relative Gap = 10^{-2}

(b) Iterations, Relative Gap = 10^{-7}

(c) Evaluations, Relative Gap = 10^{-2}

(d) Evaluations, Relative Gap = 10^{-7}

Figure 5.1: Performance profiles between MONotone (black), Original Non-Monotone (blue) and Squared NonMonotone (red) and Rooted NonMonotone (green) on a set of NE problems.

- in the intermediate phase of the optimization process (after first 1000 iterations and before last 1000 ones), (5.16) is satisfied in the majority of the iterations (in most of the executions more than 80%).

Note that giving a formal definition of initial, intermediate and late phases of the optimization procedure is very hard. In this application a threshold of 1000 iterations has been a natural choice. Moreover, changing this value with another reasonable choice, would not change conclusions that might be drawn from the observations below.

The algorithms have been compared using *performance profiles* [30] on the following two measures: total number of iterations (Figure 5.1(a) and

5.1(b)) and total number of evaluations of the objective function (Figure 5.1(c) and 5.1(d)). All the executions have been stopped using 2 different values of the *relative gap*[2]:

- 10^{-2}: a very rough solution (Figure 5.1(a) and 5.1(c));
- 10^{-7}: an accurate solution (Figure 5.1(b) and 5.1(d)).

First thing to notice from Figures 5.1(a)-5.1(d) is that all the nonmonotone version of g are outperforming the monotone counter-part (MON) on both the measures. Second observation that can be pointed out is the fact that SNM is always outperforming both WNM and ONM. This make us hypothesize that for solving a NE problem efficiently with the above algorithm there is always a need for an high amount of *nonmonotonicity* in all the phases of the optimization procedure.

As anticipated, it is hard to formally define a separation between phases. For this reason Figures 5.1(a) and 5.1(c) are focusing on both initial and intermediate phases[3]. Figures 5.1(b) and 5.1(d) are instead focusing on the whole optimization procedure. Now, since the gap between SNM and ONM is way less prominent in Figures 5.1(a) and 5.1(c) w.r.t. that of Figures 5.1(b) and 5.1(d), it is not clear if an high amount of *nonmonotonicity* is also really needed in the initial phase. From the gap between SNM and WNM, it is safer to conclude that even in the initial phase the degree of *nonmonotonicity* should not be too low. On the other hand, from Figures 5.1(a)-5.1(d) and especially from Figures 5.1(b) and 5.1(d) we can conclude that a consistent amount of *nonmonotonicity* in the intermediate and late phase is helpful for solving a NE problem efficiently.

Finally it is interesting to underline that by only modifying W on ONM it was never possible to obtain results as good as those achieved by SNM. In particular in the late phase of the optimization procedure the objective function decreases very slowly. For this reason it is almost unpractical to find a nonmonotone window W (even if adaptive) that let us reach an f^k_{max} which has enough *nonmonotonicity* as the one easily achieved by SNM. All the techniques defined in [79, 80, 100] are affected by this issue.

On the other hand, designing an $\{\epsilon^k\}$ sequence which has the right degree of *nonmonotonicity* in the late phase of the optimization procedure is another

[2] A well-known measure of convergence in NE problems (see [29]).

[3] Note that the relative gap in x^0 is always around 10^{-1} and the total number of iterations needed to reach a relative gap of 10^{-2} is way more than 1000.

very hard task. In addition, even if we can find an $\{\epsilon^k\}$ sequence which is working for one network, it is hard that the same $\{\epsilon^k\}$ is also helpful for some others. Networks are in fact very different in dimensions (see Table 5.1) and it is very common that the amount of *nonmonotonicity* designed for one network would not fit the nonmonotone effect required for others. The proposed framework (5.15) is instead self adapting to each single run and to the precision needed in each different phases. To design such a family we just need to understand the amount of *nonmonotonicity* required in the different phases. As a buy product, this study would also give us some useful insights on the preferred paths of the optimization procedure on this class of problems.

5.5 Conclusions

In this work it was first identified a significant connection between two very popular nonmonotone techniques: the original one from [55] and the more recent one from [70]. This connection was exploited to develop a new family of nonmonotone techniques that is able to obtain the best from both worlds. In particular it is easy to be designed (as [55]) and has strong control over the degree of nonmonotonicity introduced (as [70]). Moreover, the new family directly inherits global convergence properties from [70], while numerically it self adapts to each problem/phase/method on which is applied to (like [55]). Numerical results on a set of large network equilibrium problems support the thesis that controlling the amount of nonmonotonicity can be very helpful to improve performances (even beyond benefits obtained by the original [55]). In conclusions, these results show how the new general nonmonotone technique can be easily implemented on a very specific application. A possible future research would be that of identifying a wide set of other classes of problems in which nonmonotonicity has been shown to be helpful. Once obtained this set of problems, it would be interesting to apply the new tool on each of them to obtain a systematic study on the amount of nonmonotonicity actually needed.

Chapter 6

Conclusion

In this thesis, nonmonotone techniques have been successfully applied to both trust-region and line search globalization strategies. Original monotone methods have been extended to the nonmonotone case and new global convergence results have been proved under the same original conditions. Numerical results always showed the supremacy of the nonmonotone version over their monotone counter-part, both on GNEPs and NE problems. A smart implementation was designed for large-scale problems to maintain the nonmonotone effect even in case of cancellation errors. Moreover, a new family of nonmonotone techniques pushed even further benefits already obtained by switching from a monotone condition to a nonmonotone one.

As a future research our aim is that of synthesizing a practical rule for letting researchers apply nonmonotone techniques with awareness and a deeper understanding of this strategy. To do so, we will first focus on a systematic literature review on the most important papers that exploit a nonmonotone technique. In addition, we will apply the last developed nonmonotone family to a wider range of problems to understand if it is possible to extract some common patterns of best-practice. Finally, as a special case, we will focus on the application of nonmonotone techniques on optimization methods exploited in state-of-art machine learning softwares, like truncated Newton methods.

Appendix A

Publications

This research activity has led to several publications in international journals. These are summarized below.[1]

International Journals

1. **L. Galli**, C. Kanzow, M. Sciandrone. "A nonmonotone trust-region method for generalized Nash equilibrium and related problems with strong convergence properties", *Computational Optimization and Applications*, vol. 69, iss. 3, pp. 629–652, 2018. [DOI:10.1007/s10589-017-9960-3]
2. **L. Galli**, A. Galligari, M. Sciandrone. "A Unified Convergence Framework for Nonmonotone Inexact Decomposition Methods", *Computational Optimization and Applications, vol. 75, iss. 1, pp. 113-144*, 2019.

Submitted

1. **L. Galli**, "Controlling the degree of nonmonotonicity: a new line search framework combining two nonmonotone techniques", *Operational Research Letters*, 2019.

[1] The author's bibliometric indices are the following: H-index = 1, total number of citations = 2 (source: Scopus on Month October, 2019).

Bibliography

[1] M. Ahookhosh and K. Amini, “A nonmonotone trust region method with adaptive radius for unconstrained optimization problems,” *Computers and Mathematics with Applications*, vol. 60, no. 3, pp. 411–422, 2010.

[2] M. Ahookhosh, K. Amini, and M. R. Peyghami, “A nonmonotone trust-region line search method for large-scale unconstrained optimization,” *Applied Mathematical Modelling*, vol. 36, no. 1, pp. 478–487, 2012.

[3] M. Ahookhosh and S. Ghaderi, “On efficiency of nonmonotone armijo-type line searches,” *Applied Mathematical Modelling*, vol. 43, pp. 170–190, 2017.

[4] K. Amini, M. Ahookhosh, and H. Nosratipour, “An inexact line search approach using modified nonmonotone strategy for unconstrained optimization,” *Numerical Algorithms*, vol. 66, no. 1, pp. 49–78, 2014.

[5] L. Armijo, “Minimization of functions having Lipschitz continuous first partial derivatives,” *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

[6] R. Barr and E. Gilbert, “Some efficient algorithms for a class of abstract optimization problems arising in optimal control,” *IEEE Transactions on Automatic Control*, vol. 14, no. 6, pp. 640–652, 1969.

[7] S. Bellavia, M. Macconi, and B. Morini, “An affine scaling trust-region approach to bound-constrained nonlinear systems,” *Applied Numerical Mathematics*, vol. 44, no. 3, pp. 257–280, 2003.

[8] ——, “STRSCNE: A scaled trust-region solver for constrained nonlinear equations,” *Computational Optimization and Applications*, vol. 28, no. 1, pp. 31–50, 2004.

[9] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods.* Prentice Hall Englewood Cliffs, NJ, 1989, vol. 23.

[10] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA 02178-9998: Athena Scientific, 1999.

[11] E. G. Birgin, J. M. Martínez, and M. Raydan, “Nonmonotone spectral projected gradient methods on convex sets,” *SIAM Journal on Optimization*, vol. 10, no. 4, pp. 1196–1211, 2000.

[12] E. Birgin, J. M. Martínez, and M. Raydan, “Inexact spectral projected gradient methods on convex sets,” *IMA Journal of Numerical Analysis*, vol. 23, no. 4, pp. 539–559, 2003.

[13] I. Bomze, F. Rinaldi, and S. R. Buló, “First-order methods for the impatient: support identification in finite time with convergent Frank-Wolfe variants,” *SIAM Journal on Optimization*, In press.

[14] S. Bonettini, “Inexact block coordinate descent methods with application to non-negative matrix factorization,” *IMA Journal of Numerical Analysis*, vol. 31, no. 4, pp. 1431–1452, 2011.

[15] C. Buzzi, L. Grippo, and M. Sciandrone, “Convergent decomposition techniques for training RBF neural networks,” *Neural Computation*, vol. 13, no. 8, pp. 1891–1920, 2001.

[16] A. Cassioli, D. Di Lorenzo, and M. Sciandrone, “On the convergence of inexact block coordinate descent methods for constrained optimization,” *European Journal of Operational Research*, vol. 231, no. 2, pp. 274–281, 2013.

[17] R. M. Chamberlain, M. J. D. Powell, C. Lemarechal, and H. C. Pedersen, “The watchdog technique for forcing convergence in algorithms for constrained optimization,” in *Algorithms for Constrained Minimization of Smooth Nonlinear Functions*. Springer, 1982, pp. 1–17.

[18] S. C. Dafermos and F. T. Sparrow, “The traffic assignment problem for a general network,” *Journal of Research of the National Bureau of Standards B*, vol. 73, no. 2, pp. 91–118, 1969.

[19] Y.-H. Dai, “On the nonmonotone line search,” *Journal of Optimization Theory and Applications*, vol. 112, no. 2, pp. 315–330, 2002.

[20] Y.-H. Dai and R. Fletcher, “Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming,” *Numerische Mathematik*, vol. 100, no. 1, pp. 21–47, 2005.

[21] ——, “New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds,” *Mathematical Programming*, vol. 106, no. 3, pp. 403–421, 2006.

[22] Y.-H. Dai, W. W. Hager, K. Schittkowski, and H. Zhang, “The cyclic Barzilai-Borwein method for unconstrained optimization,” *IMA Journal of Numerical Analysis*, vol. 26, no. 3, pp. 604–627, 2006.

[23] Y.-H. Dai, J. Yuan, and Y.-X. Yuan, “Modified two-point stepsize gradient methods for unconstrained optimization,” *Computational Optimization and Applications*, vol. 22, no. 1, pp. 103–109, 2002.

[24] Y.-H. Dai and H. Zhang, "Adaptive two-point stepsize gradient algorithm," *Numerical Algorithms*, vol. 27, no. 4, pp. 377–385, 2001.

[25] H. Dan, N. Yamashita, and M. Fukushima, "Convergence properties of the inexact Levenberg-Marquardt method under local error bound conditions," *Optimization Methods and Software*, vol. 17, no. 4, pp. 605–626, 2002.

[26] R. De Leone, M. Gaudioso, and L. Grippo, "Stopping criteria for linesearch methods without derivatives," *Mathematical Programming*, vol. 30, no. 3, pp. 285–300, 1984.

[27] T. De Luca, F. Facchinei, and C. Kanzow, "A semismooth equation approach to the solution of nonlinear complementarity problems," *Mathematical Programming, Series B*, vol. 75, no. 3, pp. 407–439, 1996.

[28] N. Y. Deng, Y. Xiao, and F. J. Zhou, "Nonmonotonic trust region algorithm," *Journal of Optimization Theory and Applications*, vol. 76, no. 2, pp. 259–285, 1993.

[29] D. Di Lorenzo, A. Galligari, and M. Sciandrone, "A convergent and efficient decomposition method for the traffic assignment problem," *Computational Optimization and Applications*, vol. 60, no. 1, pp. 151–170, 2015.

[30] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

[31] A. Dreves, F. Facchinei, A. Fischer, and M. Herrich, "A new error bound result for generalized Nash equilibrium problems and its algorithmic application," *Computational Optimization and Applications*, vol. 59, no. 1-2, pp. 63–84, 2014.

[32] A. Dreves, F. Facchinei, C. Kanzow, and S. Sagratella, "On the solution of the KKT conditions of generalized Nash equilibrium problems," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 1082–1108, 2011.

[33] F. Facchinei and C. Kanzow, "A nonsmooth inexact Newton method for the solution of large-scale nonlinear complementarity problems," *Mathematical Programming, Series B*, vol. 76, no. 3, pp. 493–512, 1997.

[34] F. Facchinei, S. Lucidi, and L. Palagi, "A truncated Newton algorithm for large scale box constrained optimization," *SIAM Journal on Optimization*, vol. 12, no. 4, pp. 1100–1125, 2002.

[35] F. Facchinei and C. Kanzow, "Generalized Nash equilibrium problems," *Annals of Operations Research*, vol. 175, no. 1, pp. 177–211, 2010.

[36] F. Facchinei, C. Kanzow, S. Karl, and S. Sagratella, "The semismooth Newton method for the solution of quasi-variational inequalities," *Computational Optimization and Applications*, vol. 62, no. 1, pp. 85–109, 2015.

[37] F. Facchinei, C. Kanzow, and S. Sagratella, *QVILIB: A library of quasi-variational inequality test problems.* Inst. of Math., 2012.

[38] ——, "Solving quasi-variational inequalities via their KKT conditions," *Mathematical Programming*, vol. 144, no. 1-2, pp. 369–412, 2014.

[39] J.-Y. Fan and Y.-X. Yuan, "On the quadratic convergence of the Levenberg-Marquardt method without nonsingularity assumption," *Computing*, vol. 74, no. 1, pp. 23–39, 2005.

[40] M. C. Ferris and S. Lucidi, "Nonmonotone stabilization methods for nonlinear equations," *Journal of Optimization Theory and Applications*, vol. 81, no. 1, pp. 53–71, 1994.

[41] M. C. Ferris, S. Lucidi, and M. Roma, "Nonmonotone curvilinear line search methods for unconstrained optimization," *Computational Optimization and Applications*, vol. 6, no. 2, pp. 117–136, 1996.

[42] R. Fletcher, "A limited memory steepest descent method," *Mathematical Programming*, vol. 135, no. 1-2, pp. 413–436, 2012.

[43] ——, "On the Barzilai-Borwein method," in *Optimization and Control with Applications.* Springer, 2005, pp. 235–256.

[44] J. Fu and W. Sun, "Nonmonotone adaptive trust-region method for unconstrained optimization problems," *Applied Mathematics and Computation*, vol. 163, no. 1, pp. 489–504, 2005.

[45] L. Galli, A. Galligari, and M. Sciandrone, "A unified convergence framework for nonmonotone inexact decomposition methods," *Computational Optimization and Applications*, vol. 75, no. 1, pp. 113–144, 2019.

[46] L. Galli, C. Kanzow, and M. Sciandrone, "A nonmonotone trust-region method for generalized Nash equilibrium and related problems with strong convergence properties," *Computational Optimization and Applications*, vol. 69, no. 3, pp. 629–652, 2018.

[47] R. García, A. Marín, and M. Patriksson, "Column generation algorithms for nonlinear optimization, i: Convergence analysis," *Optimization*, vol. 52, no. 2, pp. 171–200, 2003.

[48] A. Griewank and P. L. Toint, "Local convergence analysis for partitioned quasi-Newton updates," *Numerische Mathematik*, vol. 39, no. 3, pp. 429–448, 1982.

[49] ——, "Partitioned variable metric updates for large structured optimization problems," *Numerische Mathematik*, vol. 39, no. 1, pp. 119–137, 1982.

[50] L. Grippo, F. Lampariello, and S. Lucidi, "A truncated Newton method with nonmonotone line search for unconstrained optimization," *Journal of Optimization Theory and Applications*, vol. 60, no. 3, pp. 401–419, 1989.

[51] ——, “A class of nonmonotone stabilization methods in unconstrained optimization,” *Numerische Mathematik*, vol. 59, no. 1, pp. 779–805, 1991.

[52] L. Grippo and F. Rinaldi, “A class of derivative-free nonmonotone optimization algorithms employing coordinate rotations and gradient approximations,” *Computational Optimization and Applications*, vol. 60, no. 1, 2014.

[53] L. Grippo and M. Sciandrone, “Nonmonotone globalization techniques for the Barzilai-Borwein gradient method,” *Computational Optimization and Applications*, vol. 23, no. 2, pp. 143–169, 2002.

[54] ——, “Nonmonotone derivative-free methods for nonlinear equations,” *Computational Optimization and Applications*, vol. 37, no. 3, pp. 297–328, 2007.

[55] L. Grippo, F. Lampariello, and S. Lucidi, “A nonmonotone line search technique for Newton’s method,” *SIAM Journal on Numerical Analysis*, vol. 23, no. 4, pp. 707–716, 1986.

[56] L. Grippo and M. Sciandrone, “Globally convergent block-coordinate techniques for unconstrained optimization,” *Optimization Methods and Software*, vol. 10, no. 4, pp. 587–637, 1999.

[57] N.-Z. Gu and J.-T. Mo, “Incorporating nonmonotone strategies into the trust region method for unconstrained optimization,” *Computers and Mathematics with Applications*, vol. 55, no. 9, pp. 2158–2172, 2008.

[58] W. W. Hager and H. Zhang, “A new active set algorithm for box constrained optimization,” *SIAM Journal on Optimization*, vol. 17, no. 2, pp. 526–557, 2006.

[59] M. Herrich, “Local convergence of Newton-type methods for nonsmooth constrained equations and applications,” *PhD Thesis*, 2014.

[60] C.-W. Hsu and C.-J. Lin, “A simple decomposition method for support vector machines,” *Machine Learning*, vol. 46, no. 1-3, pp. 291–314, 2002.

[61] S.-L. Hu, Z.-H. Huang, and N. Lu, “A non-monotone line search algorithm for unconstrained optimization,” *Journal of Scientific Computing*, vol. 42, no. 1, pp. 38–53, 2010.

[62] A. F. Izmailov and M. V. Solodov, “On error bounds and Newton-type methods for generalized Nash equilibrium problems,” *Computational Optimization and Applications*, vol. 59, no. 1-2, pp. 201–218, 2014.

[63] C. Kanzow, “Some noninterior continuation methods for linear complementarity problems,” *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 851–868, 1996.

[64] C. Kanzow and D. Steck, “Augmented Lagrangian methods for the solution of generalized Nash equilibrium problems,” *SIAM Journal on Optimization*, vol. 26, no. 4, pp. 2034–2058, 2016.

[65] C. Kanzow, N. Yamashita, and M. Fukushima, "Levenberg-Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints," *Journal of Computational and Applied Mathematics*, vol. 172, no. 2, pp. 375–397, 2004.

[66] C. Kao, L.-F. Lee, and M. M. Pitt, "Simulated maximum likelihood estimation of the linear expenditure system with binding non-negativity constraints," *Annals of Economics and Finance*, vol. 2, no. 1, pp. 203–223, 2001.

[67] W. La Cruz, J. M. Martínez, and M. Raydan, "Spectral residual method without gradient information for solving large-scale nonlinear systems of equations," *Mathematics of Computation*, vol. 75, no. 255, pp. 1429–1448, 2006.

[68] W. La Cruz and M. Raydan, "Nonmonotone spectral methods for large-scale nonlinear systems," *Optimization Methods and Software*, vol. 18, no. 5, pp. 583–599, 2003.

[69] C. Li, W. Yin, H. Jiang, and Y. Zhang, "An efficient augmented Lagrangian method with applications to total variation minimization," *Computational Optimization and Applications*, vol. 56, no. 3, pp. 507–530, 2013.

[70] D.-H. Li and M. Fukushima, "A globally and superlinearly convergent gauss–Newton-based BFGS method for symmetric nonlinear equations," *SIAM Journal on Numerical Analysis*, vol. 37, no. 1, pp. 152–172, 1999.

[71] ——, "A derivative-free line search and global convergence of Broyden-like method for nonlinear equations," *Optimization Methods and Software*, vol. 13, no. 3, pp. 181–201, 2000.

[72] C.-J. Lin, "On the convergence of the decomposition method for support vector machines," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1288–1298, 2001.

[73] ——, "A formal analysis of stopping criteria of decomposition methods for support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1045–1052, 2002.

[74] C.-J. Lin, S. Lucidi, L. Palagi, A. Risi, and M. Sciandrone, "Decomposition algorithm model for singly linearly-constrained problems subject to lower and upper bounds," *Journal of Optimization Theory and Applications*, vol. 141, no. 1, pp. 107–126, 2009.

[75] G.-H. Liu, J.-Y. Han, and D.-F. Sun, "Global convergece of the BFGS algorithm with nonmonotone linesearch," *Optimization*, vol. 34, no. 2, pp. 147–159, 1995.

[76] J. Liu and C. Ma, "A nonmonotone trust region method with new inexact line search for unconstrained optimization," *Numerical Algorithms*, vol. 64, no. 1, pp. 1–20, 2013.

[77] S. Lucidi, F. Rochetich, and M. Roma, "Curvilinear stabilization techniques for truncated Newton methods in large scale unconstrained optimization," *SIAM Journal on Optimization*, vol. 8, no. 4, pp. 916–939, 1998.

[78] J. Mo, C. Liu, and S. Yan, "A nonmonotone trust region method based on nonincreasing technique of weighted average of the successive function values," *Journal of Computational and Applied Mathematics*, vol. 209, no. 1, pp. 97–108, 2007.

[79] H. Nosratipour, A. H. Borzabadi, and O. S. Fard, "On the nonmonotonicity degree of nonmonotone line searches," *Calcolo*, vol. 54, no. 4, pp. 1217–1242, 2017.

[80] V. P. Plagianakos, G. D. Magoulas, and M. N. Vrahatis, "Deterministic nonmonotone strategies for effective training of multilayer perceptrons," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1268–1284, 2002.

[81] L. Qi, X. J. Tong, and D. H. Li, "Active-set projected trust-region algorithm for box-constrained nonsmooth equations," *Journal of Optimization Theory and Applications*, vol. 120, no. 3, pp. 601–625, 2004.

[82] M. Raydan, "The Barzilaiai and Borwein gradient method for the large scale unconstrained minimization problem," *SIAM Journal on Optimization*, vol. 7, no. 1, pp. 26–33, 1997.

[83] T. Serafini, G. Zanghirati, and L. Zanni, "Gradient projection methods for quadratic programs and applications in training support vector machines," *Optimization Methods and Software*, vol. 20, no. 2-3, pp. 347–372, 2005.

[84] Z.-J. Shi and J. Guo, "A new family of conjugate gradient methods," *Journal of Computational and Applied Mathematics*, vol. 224, no. 1, pp. 444–457, 2009.

[85] Z.-J. Shi and J. Shen, "Convergence of nonmonotone line search method," *Journal of Computational and Applied Mathematics*, vol. 193, no. 2, pp. 397–412, 2006.

[86] K. Su and D. Pu, "A nonmonotone filter trust region method for nonlinear constrained optimization," *Journal of Computational and Applied Mathematics*, vol. 223, no. 1, pp. 230–239, 2009.

[87] W. Sun, "Nonmonotone trust region method for solving optimization problems," *Applied Mathematics and Computation*, vol. 156, no. 1, pp. 159–174, 2004.

[88] W. Sun, J. Han, and J. Sun, "Global convergence of nonmonotone descent methods for unconstrained optimization problems," *Journal of Computational and Applied Mathematics*, vol. 146, no. 1, pp. 89–98, 2002.

[89] W.-Y. Sun and Q.-Y. Zhou, "An unconstrained optimization method using nonmonotone second order goldstein's line search," *Science in China, Series A: Mathematics*, vol. 50, no. 10, pp. 1389–1400, 2007.

[90] P. L. Toint, "An assessment of nonmonotone linesearch techniques for unconstrained optimization," *SIAM Journal on Scientific Computing*, vol. 17, no. 3, pp. 725–739, 1996.

[91] ——, "Non-monotone trust-region algorithms for nonlinear optimization subject to convex constraints," *Mathematical Programming*, vol. 77, no. 3, pp. 69–94, 1997.

[92] X. J. Tong and L. Qi, "On the convergence of a trust-region method for solving constrained nonlinear equations with degenerate solutions," *Journal of Optimization Theory and Applications*, vol. 123, no. 1, pp. 187–211, 2004.

[93] P. Tseng and S. Yun, "A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training," *Computational Optimization and Applications*, vol. 47, no. 2, pp. 179–206, 2010.

[94] M. Ulbrich, "Nonmonotone trust-region methods for bound-constrained semismooth equations with applications to nonlinear mixed complementarity problems," *SIAM Journal on Optimization*, vol. 11, no. 4, pp. 889–917, 2001.

[95] N. Yamashita and M. Fukushima, "On the rate of convergence of the Levenberg-Marquardt method," in *Topics in numerical analysis.* Springer, 2001, pp. 239–249.

[96] Y. H. Yu and L. Gao, "Nonmonotone line search algorithm for constrained minimax problems," *Journal of Optimization Theory and Applications*, vol. 115, no. 2, pp. 419–446, 2002.

[97] Z. Yu, J. Lin, J. Sun, Y. Xiao, L. Liu, and Z. Li, "Spectral gradient projection method for monotone nonlinear equations with convex constraints," *Applied Numerical Mathematics*, vol. 59, no. 10, pp. 2416–2423, 2009.

[98] Z. Yu and D. Pu, "A new nonmonotone line search technique for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 219, no. 1, pp. 134–144, 2008.

[99] L. Zanni, "An improved gradient projection-based decomposition technique for support vector machines," *Computational Management Science*, vol. 3, no. 2, pp. 131–145, 2006.

[100] H. Zhang and W. Hager, “A nonmonotone line search technique and its application to unconstrained optimization,” *SIAM Journal on Optimization*, vol. 14, no. 4, pp. 1043–1056, 2004.

[101] J.-L. Zhang and X.-S. Zhang, “A modified SQP method with nonmonotone linesearch technique,” *Journal of Global Optimization*, vol. 21, no. 2, pp. 201–218, 2001.

[102] ——, “A nonmonotone adaptive trust region method and its convergence,” *Computers and Mathematics with Applications*, vol. 45, no. 10-11, pp. 1469–1477, 2003.

[103] J. Z. Zhang and L. H. Chen, “Nonmonotone Levenberg-Marquardt algorithms and their convergence analysis,” *Journal of Optimization Theory and Applications*, vol. 92, no. 2, pp. 393–418, 1997.

[104] Y. Zhang and Z.-H. Huang, “A nonmonotone smoothing-type algorithm for solving a system of equalities and inequalities,” *Journal of Computational and Applied Mathematics*, vol. 233, no. 9, pp. 2312–2321, 2010.

[105] J. L. Zhou and A. L. Tits, “Nonmonotone line search for minimax problems,” *Journal of Optimization Theory and Applications*, vol. 76, no. 3, pp. 455–476, 1993.

[106] W. Zhou, “A short note on the global convergence of the unmodified PRP method,” *Optimization Letters*, vol. 7, no. 6, pp. 1367–1372, 2013.