Nordic MPS 2004

The Ninth Meeting of the Nordic Section of the
Mathematical Programming Society
October 22–23, 2004
Linköpings universitet, Norrköping, Sweden

Conference Proceedings

organized by

The Department of Science and Technology,
Linköping University, Sweden and
The Nordic Section of the
Mathematical Programming Society

Edited by

Di Yuan

Published for Nordic Section of the
Mathematical Programming Society by
Linköping University Electronic Press
Linköping, Sweden, 2004

Linköping Electronic Conference Proceedings, No. 14
Linköping University Electronic Press
Linköping, Sweden, 2004

ISBN 91-85297-29-1 (print)
ISSN 1650-3686 (print)
http://www.ep.liu.se/ecp/014/
ISSN 1650-3740 (online)

Print: UniTryck, Linköping, 2004

Table of Contents

Preface
Di Yuan v

Articles
Determining the Non-Existence of Compatibel OSPF Weights
Peter Broström and Kaj Holmberg 7

Cutting Plane Methods in Decision Analysis
Xiaosong Ding and Faiz Al-Khayyal 23

Topology Optimization of Navier–Stokes Equations
Anton Evgrafov 37

A Generating Set Search Method Exploiting Curvature and Sparsity
Lennart Frimannslund and Trond Steihaug 57

Duality in MIP. Generating Dual Price Functions Using Branch-and-Cut
Elena V. Pachkova 73

Abstracts 89

Preface

This volume contains a collection of papers and abstracts of the presentations at the Ninth Meeting of the Nordic Section of the Mathematical Programming Society (Nordic MPS ’04). The meeting was organized and hosted by the Department of Science and Technology, Campus Norrköping, Linköping University, Sweden.

Nordic MPS ’04 is a continuation of the series of conferences organized by the Nordic MPS over the years. The aim of these conferences is to present advances in theory and applications of mathematical programming and related fields, as well as to provide a forum for people working in these fields. Nordic MPS ’04 featured 27 presentations in various topics of mathematical programming. As the conference chair, I would like to thank all speakers for contributing to a very successful scientific program, and all conference participants for contributing to the pleasant and inspiring atmosphere at Nordic MPS ‘04. I would also like to thank the rest of the conference organizing committee, Henrik Andersson, Tobias Andersson, Patrik Björklund, Margareta Klang, and Iana Siomina for their contributions to the conference organization.

Finally, I would like to acknowledge ILOG for sponsoring Nordic MPS ’04 and for organizing the pre-conference workshop on ILOG optimization technologies.

Di Yuan
Conference chair

Determining the Non-Existence of Compatible OSPF Weights

Peter Broström pebro@mai.liu.se

Kaj Holmberg kahol@mai.liu.se

Department of Mathematics, Linköping university, SE-58183 LINKÖPING, Sweden

Abstract

Many telecommunication networks use the OSPF protocol (Open Shortest Path First) for deciding the routing of traffic. In such networks, each router sends traffic on all shortest paths to the destination. The links in the network are assigned weights to be used by the routers when calculating the shortest paths.

An interesting question is whether or not a set of desired routing patterns can be used in an OSPF network. We investigate this problem, and find new necessary conditions for the existence of weights making the desired patterns shortest. A polynomial algorithm that for most cases verifies the non-existence of compatible weights is presented. The algorithm also indicates which parts of the traffic patterns that are in conflict. Some computational tests of the algorithm are reported.

Key words: *Telecommunication networks, Internet Protocol, OSPF, routing, compatible weights.*

1 Introduction

In telecommunication networks using IP (Internet Protocol) and the routing protocol OSPF (Open Shortest Path First), traffic is routed on the shortest paths from each router to each destination. The routers calculate themselves the shortest paths to all possible destinations. The shortest path calculations are based on link weights set by the network operator. If the shortest path is not unique, traffic leaving a router is split equally on the leaving arcs that belong to a shortest path to the destination. This is called ECMP (the equal-cost multipath principle), and means that *all* shortest paths are used.

We will here study the problem of finding weights that give certain specified traffic patterns in a directed graph. We will also study the more important question of whether or not there exists a set of weights giving the desired traffic patterns. Traffic patterns are represented by the paths to be used. (If the paths are known, it is a simple matter to calculate the actual traffic.)

Similar problems have previously been treated as optimization problems in [4], [9], [2], and [5], and in a larger model for network design in [8]. Most of this work is done for undirected graphs and for single shortest paths. Restricting the traffic patterns to contain only single paths yields a simplification of the more general case we are treating. The usage of directed graphs (i.e. allowing different weights in different directions) is also an important generalization.

Apart from that, our main contribution is the way in which we verify and characterize the non-existence of weights. This is done by identifying combinations of traffic patterns that prohibit the existence of weights. This yields necessary conditions for the existence of weights, that are stronger than the conditions previously known. In addition, we present a polynomial method that explains why the specified patterns can not be used in an IP/OSPF network. This could be very useful in a planning process, since it identifies the parts of the patterns that need to be modified.

The paper is organized as follows. Section 2 is used for presenting the problem in detail, and for presenting a linear model which is used for finding appropriate weights. The LP-dual is formulated in section 3, and one type of unbounded solution to the LP-dual is classified in section 4. The solution method is presented in section 5, while possible modifications of SP-graphs are discussed in section 6. The method is exemplified in section 7, while computational results are presented in section 8. The last section concludes the paper and identifies parts that will be studied further.

2 Problem formulation

We consider a directed graph $G = (N, A)$ with a set of nodes N and a set of arcs A. A number of subsets of the arcs, $A_l \subseteq A$ for $l = 1, \ldots, m$, called SP-graphs (shortest path graphs), are given. We assume that each set A_l contains a spanning tree (ignoring the direction of the arcs) and that no set A_l contains a directed cycle. (These assumptions are motivated below.)

An SP-graph contains a number of paths, and these paths are the desired shortest paths. We wish to find weights w_{ij} for all arcs $(i, j) \in A$, so that the paths in A_l have minimal sum of the weights (i.e. are shortest). Thus if A_l contains a path from node s to node t, this path should have a minimal sum of weights. All paths from s to t not completely in A_l should have larger sums of weights. If A_l contains more than one path from node s to node t, all these paths should have (the same) minimal sum of weights.

Let us by $W(p)$ denote the sum of weights of all arcs in path p, i.e. $W(p) = \sum_{(i,j)\in p} w_{ij}$. If $p(s,t)$ and $q(s,t)$ are two paths from node s to node t, and $p(s,t) \subseteq A_l$ while $q(s,t) \not\subseteq A_l$, we require that $W(p(s,t)) < W(q(s,t))$. If both $p(s,t)$ and $q(s,t)$ lie in A_l, then we should have $W(p(s,t)) = W(q(s,t))$.

The case when there is at most one path in A_l between a pair of nodes is called the *simple path case*. More work has been done on the simple path case (see e.g. [4] and [5]) than on the more general case. However, in order to enable the use of *load balancing*, which is important in practice, an SP-graph must be allowed to contain several paths between a pair of nodes.

An SP-graph is meant to be the result of a router's shortest path calculations to all possible destinations, so usually an SP-graph is an out-graph with a single origin, spanning all nodes. Different SP-graphs then have different origins.

The weights w_{ij} must be integers greater or equal to 1. In principle there is an upper bound, namely the largest integer that can be represented by the router, for example 2^{16}, but this upper bound is considered to be redundant. In this context, we might mention that in [6] only weights up to 20 are considered when demonstrating the advantages of optimizing over the weights.

Definition 1 *The weights w are said to be compatible with A_l if $W(p(s,t)) = W(r(s,t))$ for any two paths $p(s,t) \subseteq A_l$ and $r(s,t) \subseteq A_l$, and $W(p(s,t)) < W(q(s,t))$ for any two paths $p(s,t) \subseteq A_l$ and $q(s,t) \not\subseteq A_l$.*

We will use the term **compatible weights** for a set of weights w that are compatible with each A_l for $l = 1, \ldots, m$. This means that compatible weights simultaneously give all the desired shortest paths for all SP-graphs. Our first objective is to *find a set of compatible weights.*

Note that the weights do not depend on l, so the existence of compatible weights imply some sort of similarities between the sets A_l. Given the sets A_l for $l = 1, \ldots, m$, there are two possibilities, either compatible weights exist or they don't. If compatible weights exist, the difference between two different sets of compatible weights is often unimportant. Thus we will mainly address the following question: *Does a set of compatible weights exist?*

We are not only interested in the yes/no answer to this question. If the answer is yes, we wish to find compatible weights. More importantly, if the answer is no, we wish to identify the parts of the SP-graphs that prohibit the existence of compatible weights. This will open possibilities of modifying SP-graphs so that compatible weights can be found.

If the weights w are given, the routers determine the shortest paths to each destination by solving shortest path problems. The SP-graphs have different origins/destinations, so we need to solve one shortest path problem for each SP-graph l.

Let $P^l(w)$ denote the shortest path problem obtained for the weights w and the origins/destinations given by SP-graph l. If we let b^l denote the right-hand-side of the constraints of $P^l(w)$, then the differences between SP-graphs are restricted to b^l. The LP-dual of the shortest path problem $P^l(w)$ is given below.

$$\max \sum_i b_i^l y_i \text{ s.t. } -y_i + y_j \leq w_{ij} \; \forall (i,j)$$

Here the objective function depends on l, while the feasible set does not. The dual constraints state $w_{ij} + y_i - y_j \geq 0 \; \forall (i,j)$, while the complementary slackness conditions tell us that only arcs with $w_{ij} + y_i - y_j = 0$ can be used by the shortest paths.

Since a shortest path problem is an LP-problem, an optimal solution is a basic solution, and thus forms a spanning tree. This means that there is a spanning tree of arcs with $w_{ij} + y_i - y_j = 0$. If the shortest paths are not unique, there are additional arcs with $w_{ij} + y_i - y_j = 0$. The arcs with $w_{ij} + y_i - y_j = 0$ will however never form a directed cycle, since the weights are positive.

A first necessary condition for the existence of compatible weights is that no SP-graph contains a conflict in itself. In other words, we assume that there exists a set of weights compatible with each SP-graph A_l. Disagreement only occurs as conflicts between two or more SP-graphs. We assume that each SP-graph has been obtained by solving a shortest path problem (for some given weights). This motivates the assumtions that each SP-graph spans all nodes and that no SP-graph contains a directed cycle.

Let us now construct a mathematical model for the problem of finding compatible weights. The problem is really only a feasibility problem, but let us add the goal of minimizing the sum of the weights. This is no important goal, but there is no point in letting the weights become unnecessarily large.

$$\begin{array}{lrcll} \min & \displaystyle\sum_{(i,j)\in A} w_{ij} & & & & \\ \text{s.t.} & w_{ij} + \pi_i^l - \pi_j^l & = & 0 \;\; \forall (i,j) \in A_l, l = 1, \ldots, m & (1.1) \\ & w_{ij} + \pi_i^l - \pi_j^l & \geq & 1 \;\; \forall (i,j) \notin A_l, l = 1, \ldots, m & (1.2) \\ & w_{ij} & \geq & 1 \;\; \forall (i,j) \in A & (1.3) \end{array} \qquad \text{(P1)}$$

Constraints 1.1 ensure that the arcs in A_l are in minimal weight paths, while constraints 1.2 ensure that arcs outside of A_l are not in minimal weight paths. The coefficients in the

objective function are unimportant, and could be replaced by any nonnegative coefficients.

The solution of P1 will be rational, and by multiplying w and π by a constant of appropriate size, it can be made integral. It is easy to show that this retains feasibility in P1. (For details, see [3].)

Lemma 1 *If there exists a feasible solution to P1, there exists a feasible integer solution to P1.*

For a certain l each arc appears once in constraint set 1.1 or once in constraint set 1.2, since an arc is either in the SP-graph or not in it. Therefore any feasible solution to P1 will satisfy $w_{ij} + \pi_i^l - \pi_j^l \geq 0 \ \forall (i,j), \forall l$. Summing these constraints over any path $p(s,t)$ yields

$$W(p(s,t)) = \sum_{(i,j)\in p(s,t)} w_{ij} \geq \sum_{(i,j)\in p(s,t)} (\pi_j^l - \pi_i^l) = \pi_t^l - \pi_s^l,$$

so we have

$$W(p(s,t)) \geq \pi_t^l - \pi_s^l \qquad \text{for any path } p(s,t) \text{ and any } l. \tag{1.4}$$

The number of variables in P1 is $|A| + m|N|$, and the number of constraints is equal to $(m+1)|A|$, so the size of P1 is quite reasonable.

Theorem 1 *P1 has a feasible solution if and only if there exists a set of compatible weights.*

Proof: Consider a certain SP-graph A_l and two nodes s and t such that there exists a path $p(s,t)$ in A_l from node s to node t. Now assume that P1 has a feasible solution, w and π. The sum of weights for a path $p(s,t) \subseteq A_l$ is

$$W(p(s,t)) = \sum_{(i,j)\in p(s,t)} w_{ij} = \sum_{(i,j)\in p(s,t)} (\pi_j^l - \pi_i^l) = \pi_t^l - \pi_s^l,$$

due to constraints 1.1. Let d_{st} be the minimal sum of weights w on any path from node s to node t. Since $p(s,t)$ is one possible path from s to t, we have $W(p(s,t)) \geq d_{st}$, so $d_{st} \leq \pi_t^l - \pi_s^l$.

There must exist a path $q(s,t)$ from node s to node t with minimal sum of weights. For such a path we get $d_{st} = W(q(s,t)) \geq \pi_t^l - \pi_s^l$ due to 1.4, i.e. $d_{st} \geq \pi_t^l - \pi_s^l$.

Combining these results yields $d_{st} = \pi_t^l - \pi_s^l$. Above we noted that if $p(s,t) \subseteq A_l$ then $W(p(s,t)) = \pi_t^l - \pi_s^l$, so then $W(p(s,t)) = d_{st}$, which means that $p(s,t)$ is a minimal weight path. We have thus proved that any path in A_l is a minimal weight path.

Now consider another path $r(s,t) \not\subseteq A_l$ between the same two nodes. Constraints 1.1 and 1.2 yields the following.

$$W(r(s,t)) = \sum_{(i,j)\in r(s,t)} w_{ij} = \sum_{(i,j)\in r(s,t)\cap A_l} w_{ij} + \sum_{(i,j)\in r(s,t)\setminus A_l} w_{ij} \geq \sum_{(i,j)\in r(s,t)\cap A_l} (\pi_j^l - \pi_i^l) + \sum_{(i,j)\in r(s,t)\setminus A_l} (\pi_j^l - \pi_i^l + 1) = \sum_{(i,j)\in r(s,t)} (\pi_j^l - \pi_i^l) + |r(s,t) \setminus A_l| = \pi_t^l - \pi_s^l + |r(s,t) \setminus A_l| = d_{st} + |r(s,t) \setminus A_l| > d_{st}.$$

This shows that any path with $|r(s,t) \setminus A_l| > 0$ (i.e. at least one arc outside of A_l) has $W(r(s,t)) > d_{st}$, i.e. is not a minimal weight path. So for any l, we have shown that the paths in A_l are minimal weight paths, and paths not completely in A_l are not minimal weight paths. This verifies that the weights w are compatible with all SP-graphs, so there exists a compatible set of weights if P1 has a feasible solution.

Let us now assume that there exists a set of compatible weights, $w \geq 1$. Then we can solve the shortest path problems $P^l(w)$ for each l, and get the dual solutions y^l. Since the weights are compatible, all arcs in the SP-graphs will be included in minimal weight paths, while arcs outside of the SP-graphs will not.

This means that $w_{ij} + y_i^l - y_j^l = 0$ for each arc in A_l, while $w_{ij} + y_i^l - y_j^l > 0$ for each arc outside of A_l. If $w_{ij} + y_i^l - y_j^l < 1$ for some $(i,j) \notin A_l$, then w and π can be multiplied with a positive constant of appropriate size, in order to make the solution satisfy constraints 1.2. (See also lemma 1.) This verifies that there exists a feasible solution to P1 if there exists a compatible set of weights. □

Comments: It may be noted that that in the first part of the proof, no additional assumptions were made on the structure of the SP-graphs. In the second part, however, we note that shortest path problems yield spanning shortest path trees, so if the SP-graphs were not spanning, not connected or contained directed cycles, constraints 1.2 might not be satisfied.

In conclusion, if P1 has a feasible solution, no further assumptions on the SP-graphs are necessary. Verifying that P1 has a feasible solution if comptaible weights exist, however, requires that each SP-graph can be obtained by solving a shortest path problem.

It should be pointed out that nothing in the proof prohibits SP-graphs from containing several paths between a pair of nodes.

3 Using LP-duality

As mentioned above, our main interest lies in whether or not there exists compatible weights. This question can now be reformulated to whether or not P1 has a feasible solution. We start by formulating the LP-dual to P1.

Let γ_{ij}^l be the dual variables to constraint sets 1.1 and 1.2 (note that for any l, each arc appears once, either in constraint set 1.1 or in constraint set 1.2), and let δ_{ij} be the dual variables for constraint set 1.3. The LP-dual can be formulated as follows.

$$
\begin{aligned}
\max \quad & \sum_{l=1}^{m} \sum_{(i,j)\notin A_l} \gamma_{ij}^l + \sum_{(i,j)\in A} \delta_{ij} & & & \\
\text{s.t.} \quad & \sum_{l=1}^{m} \gamma_{ij}^l + \delta_{ij} & = & 1 \quad \forall (i,j) \in A & (2.1) \\
& \sum_{j:(i,j)\in A} \gamma_{ij}^l - \sum_{j:(j,i)\in A} \gamma_{ji}^l & = & 0 \quad \forall i \in N, l = 1,\ldots,m & (2.2) \\
& \delta_{ij} & \geq & 0 \quad \forall (i,j) \in A & (2.3) \\
& \gamma_{ij}^l & \geq & 0 \quad \forall (i,j) \notin A_l, l = 1,\ldots,m & (2.4)
\end{aligned}
\tag{P2}
$$

Note that γ_{ij}^l is free (not sign-restricted) for $(i,j) \in A_l, l = 1,\ldots,m$, and that these variables do not appear in the objective function.

Let us first eliminate δ. Constraint set 2.1 immediately gives $\delta_{ij} = 1 - \sum_{l=1}^m \gamma_{ij}^l$, and constraint set 2.3 yields $\sum_{l=1}^m \gamma_{ij}^l \leq 1$. Doing the substitution in the objective function yields

$$
\sum_{l=1}^{m} \sum_{(i,j)\notin A_l} \gamma_{ij}^l + \sum_{(i,j)\in A} (1 - \sum_{l=1}^{m} \gamma_{ij}^l) = |A| - \sum_{l=1}^{m} \sum_{(i,j)\in A_l} \gamma_{ij}^l.
$$

We now ignore the constant $|A|$ and change from maximization to minimization. (The actual objective function value is unimportant.) We have now simplified P2 to the following.

$$
\begin{aligned}
\min \quad & \sum_{l=1}^{m} \sum_{(i,j)\in A_l} \gamma_{ij}^l & & & & \\
\text{s.t.} \quad & \sum_{l=1}^{m} \gamma_{ij}^l & \leq & 1 & \forall (i,j) \in A & (3.1) \\
& \sum_{j:(i,j)\in A} \gamma_{ij}^l - \sum_{j:(j,i)\in A} \gamma_{ji}^l & = & 0 & \forall i \in N, l = 1, \ldots, m & (3.2) \\
& \gamma_{ij}^l & \geq & 0 & \forall (i,j) \notin A_l, l = 1, \ldots, m & (3.3)
\end{aligned}
\qquad \text{(P3)}
$$

Let v be the optimal objective function value of P3. The only difference between P3 and a standard *multicommodity network flow problem*, see for example [1], is that some variables are free (not nonnegative). Our goal at the moment is not to solve this problem computationally, but rather to investigate its properties.

Constraint set 3.2 states that the inflow to each node must equal the outflow, for each commodity. This means that we are looking for a circulating flow, as there are no sources or sinks. Constraint set 3.1 corresponds to capacity constraints, and all capacities are equal to one. (Note that these ones are the objective function coefficients for w in P1, and could, as mentioned, be other non-negative constants.)

We note that a feasible solution is obtained by setting $\gamma_{ij}^l = 0 \ \forall (i,j) \in A, \ \forall l$. This means that $\delta_{ij} = 1 \ \forall (i,j) \in A$, and a quick look at the complementary slackness conditions reveals that this corresponds to setting $w_{ij} = 1 \ \forall (i,j) \in A$. This is unlikely to be a feasible solution in P1, but since it is a feasible solution to P3, an upper bound to the optimal objective function value of P3 is zero. A better solution can be found if some of the (free) variables in the objective function can be decreased. Lemma 1 and LP-duality gives the following result (see [3] for details).

Lemma 2 *P1 has a feasible integer solution if and only of P3 has a bounded optimal solution.*

Thus there is no feasible integer solution to P1 if P3 has an unbounded solution, so we can study P3, in order to draw conclusions about the existence of compatible weights.

4 Unbounded multicommodity flow solutions

Let us now study unbounded solutions to P3. We start at a feasible solution, for example $\bar{\gamma} = 0$, and change it such that some variables go toward infinity.

Constraints 3.2 only allows circulating flow, so we must change the flow in cycles. Consider a *cycle* $C \subseteq A$, $C = F \cup B$, where F are the arcs used *forwards* (in their directions) and B are the arcs used *backwards* (against their directions). We change the flow of commodity l' in the cycle C by increasing γ_{ij}^l with the amount θ on forward arcs, and by decreasing γ_{ij}^l with the amount θ on backward arcs. To get an unbounded solution, we need to increase θ infinitely.

Constraint set 3.1 says that $\sum_{l=1}^{m} \gamma_{ij}^l \leq 1$, so if one variable in the left-hand-side is to be increased infinitely, another variable must be decreased infinitely. Specifically, if the flow of a commodity l' in arc (i,j) is increased by θ, then the flow of another commodity, l'', in that arc must be decreased by the same amount. This can be easily accommodated by using the same cycle C for commodity l'' as for commodity l', but doing the change in reversed direction. Thus we get the following change.

$$\gamma_{ij}^{l'} = \bar{\gamma}_{ij}^{l'} + \theta \quad \forall (i,j) \in F, \quad \gamma_{ij}^{l'} = \bar{\gamma}_{ij}^{l'} - \theta \quad \forall (i,j) \in B$$
$$\gamma_{ij}^{l''} = \bar{\gamma}_{ij}^{l''} - \theta \quad \forall (i,j) \in F, \quad \gamma_{ij}^{l''} = \bar{\gamma}_{ij}^{l''} + \theta \quad \forall (i,j) \in B$$

According to constraint set 3.3, $\gamma_{ij}^{l} \geq 0 \ \forall (i,j) \notin A_l \ \forall l$, some variables can not be decreased infinitely. The flows that are decreased are commodity l' in arcs B and commodity l'' in arcs F, so these variables must not appear in any non-negativity constraint. In other words, all $(i,j) \in B$ must also be included in $A_{l'}$ and all $(i,j) \in F$ must also be included in $A_{l''}$. This means that it is necessary that $B \subseteq A_{l'}$ and $F \subseteq A_{l''}$. This can also be written as $|B \cap A_{l'}| = |B|$ and $|F \cap A_{l''}| = |F|$. Furthermore, since $|B \cap A_{l'}| \leq |B|$ and $|F \cap A_{l''}| \leq |F|$, an equivalent statement is that $|B \cap A_{l'}| + |F \cap A_{l''}| = |B| + |F|$.

Lemma 3 *The flow in a cycle $C = F \cup B$ can only be increased infinitely if $B \subseteq A_{l'}$ and $F \subseteq A_{l''}$.*

We call such a cycle a *feasible* cycle, while if $|B \cap A_{l'}| + |F \cap A_{l''}| < |B| + |F|$, the cycle is called *infeasible*.

In order for P3 to have an unbounded solution, the objective function value must be decreased infinitely. Inserting the parameterized solution into the objective function yields

$$\begin{aligned} v &= \sum_{l=1}^{m} \sum_{(i,j) \in A_l} \gamma_{ij}^{l} = \sum_{l=1}^{m} \sum_{(i,j) \in A_l} \bar{\gamma}_{ij}^{l} + \sum_{(i,j) \in F \cap A_{l'}} \theta - \sum_{(i,j) \in B \cap A_{l'}} \theta - \sum_{(i,j) \in F \cap A_{l''}} \theta + \sum_{(i,j) \in B \cap A_{l''}} \theta \\ &= \Gamma + (|F \cap A_{l'}| - |B \cap A_{l'}| - |F \cap A_{l''}| + |B \cap A_{l''}|)\theta = \Gamma + \hat{r}^C \theta, \end{aligned}$$

where $\hat{r}^C = |F \cap A_{l'}| - |B \cap A_{l'}| - |F \cap A_{l''}| + |B \cap A_{l''}|$ is called the *reduced cost* for cycle C, and $\Gamma = \sum_{l=1}^{m} \sum_{(i,j) \in A_l} \bar{\gamma}_{ij}^{l}$, which is a constant.

In order for $v \to -\infty$ as $\theta \to \infty$, the reduced cost $\hat{r}^C$ must be negative. Now we remember that this unbounded solution is feasible only if $|B \cap A_{l'}| = |B|$ and $|F \cap A_{l''}| = |F|$, so the reduced cost becomes $\hat{r}^C = |F \cap A_{l'}| + |B \cap A_{l''}| - |F| - |B|$ and a negative reduced cost is obtained if $|F \cap A_{l'}| + |B \cap A_{l''}| < |F| + |B|$. Since $|F \cap A_{l'}|$ can never be larger than $|F|$ and $|B \cap A_{l''}|$ can never be larger than $|B|$, the only possibility for this inequality *not* to hold (i.e. $\hat{r}^C = 0$) is that $|F \cap A_{l'}| = |F|$ and $|B \cap A_{l''}| = |B|$. In other words, in order for the reduced cost to be negative, it is enough if there is one element in F not in $A_{l'}$ or one element in B not in $A_{l''}$.

We call arc (i,j) *eligible* if either $(i,j) \in F$ and $(i,j) \notin A_{l'}$ or $(i,j) \in B$ and $(i,j) \notin A_{l''}$. Moreover, a cycle with $|F \cap A_{l'}| + |B \cap A_{l''}| < |F| + |B|$ is called an *improving* cycle, while if $|F \cap A_{l'}| + |B \cap A_{l''}| = |F| + |B|$, the cycle is called *non-improving*.

Lemma 4 *A feasible cycle $C = F \cup B$ indicates an unbounded solution of P3 only if $|F \cap A_{l'}| + |B \cap A_{l''}| < |F| + |B|$, i.e. if there is at least one eligible arc (an arc in F not in $A_{l'}$ or in B not in $A_{l''}$).*

Summing up these conclusions, we wish to find a cycle $C = F \cup B$, and two commodities l' and l'' such that $B \subseteq A_{l'}$ and $F \subseteq A_{l''}$ while $|F \cap A_{l'}| + |B \cap A_{l''}| < |F| + |B|$, which means that the cycle is both feasible and improving.

Definition 2 *A cycle $C = F \cup B$ is called <u>valid</u> if there exist two indices l' and l'' such that $|B \cap A_{l'}| = |B|$, $|F \cap A_{l''}| = |F|$ and $|F \cap A_{l'}| + |B \cap A_{l''}| < |F| + |B|$. Equivalently, for a valid cycle, $B \subseteq A_{l'}$ and $F \subseteq A_{l''}$, while $B \nsubseteq A_{l''}$ and/or $F \nsubseteq A_{l'}$.*

Theorem 2 *If there exists a valid cycle, then there exists no compatible set of weights.*

Theorem 2 can also be proved by direct arguments in P1, not using LP-duality. By walking around the cycle, one can sum up all constraints encountered, and from this draw the conclusion. The details of this are given in [3].

A previously known necessary condition for the existence of compatible weights is that if two desirered paths use the same two nodes in the same order, then the paths between the two nodes must be identical. Paths that satisfy this are called *sub-optimal*. It is in [3] shown that if two SP-graphs contain paths that are not sub-optimal, then a valid cycle exists. Furthermore, in section 7 we give an example with sub-optimal paths where a valid cycle exists. Thus, the absence of valid cycles is a stronger necessary condition for the existence of compatible weights than sub-optimality.

One can easily show the following (for details, see [3]).

Lemma 5 *A valid cycle must contain at least three nodes and three arcs.*

If all SP-graphs are trees, there does not exist any cycle within an SP-graph, even if we disregard the arc directions. Since a feasible cycle has $B \subseteq A_{l'}$, we can draw the conclusion that $F \not\subseteq A_{l'}$ for any feasible cycle. Also, since $F \subseteq A_{l''}$, we know that $B \not\subseteq A_{l''}$. This immediately tells us that any feasible cycle is improving, i.e. valid.

Theorem 3 *If the SP-graphs $A_{l'}$ and $A_{l''}$ are trees, then any feasible cycle (i.e. with $B \subseteq A_{l'}$ and $F \subseteq A_{l''}$) is also valid.*

5 A method for finding valid cycles

Let us now consider practical ways of finding valid cycles. We wish to find a valid cycle $C = F \cup B$ and two indices l' and l'' verifying its validity, i.e. such that $|B \cap A_{l'}| = |B|$, $|F \cap A_{l''}| = |F|$ and $|F \cap A_{l'}| + |B \cap A_{l''}| < |F| + |B|$.

We enumerate all pairs of commodities, i.e. try to find a valid cycle for each $l' = 1, \ldots, m$ and $l'' = 1, \ldots, m$. There are $(m-1)^2$ possibilities, but as soon as we find a valid cycle, we stop. So assume now that l' and l'' are given.

First we note that $A_{l'} \cup A_{l''}$ must cover the cycle, so arcs not belonging to any of these sets are discarded, i.e. removed from the graph. Then, since we require $B \subseteq A_{l'}$ and $F \subseteq A_{l''}$, we label each arc in $A_{l'}$ with B, and each arc in $A_{l''}$ with F. This is called the *labeling phase*.

After this, all remaining arcs are labeled at least once. Arcs labeled only with B must belong to the set B, if they are included in the cycle, and arcs labeled only with F must belong to the set F, if they are included in the cycle. Arcs labeled with B or F are eligible, while arcs labeled with both B and F are not eligible.

The next step is to remove parts of the remaining graph that can not be a part of a valid cycle. This is called the *reduction phase*. In this stage we must obey the arc labels, so that an arc labeled F can only be used forwards, and an arc labeled B can only be used backwards, and an arc labeled both with B and F can be used in both directions. For example, an arc "entering" a node i can be an arc ending in node i with label F, an arc starting in node i with label B, or an arc labeled with both F and B (regardless of original direction).

The reduction phase aims at ensuring that all remaining nodes have at least one entering arc and at least one leaving arc (different from the entering arc).

- For any node with only one adjacent arc: Discard the node and the arc.

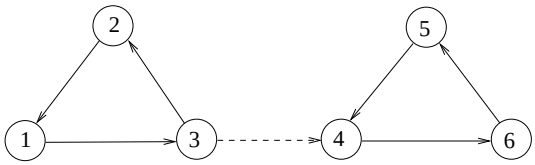


Figure 1: A graph with an arc not included in any cycle.

- If there are no arcs entering (leaving) a node: Discard the node and all adjacent arcs.
- If there is only one arc entering (leaving) a node, and this arc is labeled with both B and F: Keep the label that enables entering (leaving) the node, and remove the other label.

Furthermore we note that there must be at least three nodes and three arcs in a valid cycle, see lemma 5, so any graph smaller than that may be immediately eliminated.

- For any connected graph component with less than three nodes or less than three arcs: Discard all nodes and all arcs in the component.

These graph reductions should be repeated until no more changes occur. We need to investigate each node at least once, checking all its adjacent arcs. As soon as a change is made in the graph, all affected nodes have to be checked again. As the changes are either removing an arc label or removing a node, at most $|N| + |A|$ changes can be made. If we keep a list of the in-degree and out-degree of every node, this list can easily be updated when changes are made, especially if the change only affects one arc. An simple estimation of the complexity of the reduction phase is $O(|N|^3)$.

Lemma 6 *If the graph is completely eliminated by the reduction phase, there exists no valid cycle with the two SP-graphs considered.*

If the graph is not completely eliminated, a graph with at least three nodes remains, and it contains at least one feasible cycle. According to lemma 4, a feasible cycle is valid if it contains an eligible arc, i.e. one arc in F outside of $A_{l'}$, or one arc in B outside of $A_{l''}$. If there are no eligible arcs in the graph, there is no valid cycle in the graph, and we are finished with this commodity pair.

Let us start with an eligible arc. A cycle is found simply by traversing nodes, using a leaving arc that is different from the entering. After at most $|N|$ steps, we will return to a node previously visited, and have thus found a cycle.

It is not certain that our starting arc is included in the cycle we find. Actually, it is not certain that there exists a cycle including our starting arc. See the graph in figure 1, which might be the result of the reduction phase. Note that the graph has two cycles, and that regardless of which arc we start with, one of them will be found. However, if we start with arc (3,4), and hope to find a cycle with it in, we will fail. We will instead find the cycle 4-6-5.

In general there are two possibilities. Either there exists a cycle containing an eligible arc, or there does not exist one. In the first case, we wish to find the cycle. In the second case, we wish to verify that there is no such cycle, and conclude that there exists no valid cycle for this pair of commodities. We try to make the cycle as large as possible, since that might increase the probability that our starting arc will be included in the cycle. If

there is more than one outgoing arc from a node, we avoid if possible arcs going to already visited nodes.

As soon as we find a cycle including our starting arc or another eligible arc, we have succeeded in finding a valid cycle, and can terminate the procedure. If there is no eligible arc in the cycle, we have found a cycle with $F \subseteq A_{l'}$ and $B \subseteq A_{l''}$, i.e. a cycle with reduced cost equal to zero. Such cycles are not valid, and should if possible be removed from the graph.

If the cycle is "isolated", in the sense that the total out-degree from the cycle is equal to zero (see cycle 4-6-5 in figure 1) or the total in-degree into the cycle is equal to zero (see cycle 1-3-2 in figure 1), it can be eliminated. The reason is that the arcs in the cycle can not be a part of another cycle, since either we can not leave the cycle once we are in it, or we can not enter it from nodes outside the cycle. This reasoning can be extended to any subgraph as follows.

Definition 3 *A subgraph containing no eligible arcs and with either total in-degree equal to zero or total out-degree equal to zero, is called an isolated subgraph.*

Lemma 7 *No node in an isolated subgraph can be a part of a valid cycle.*

All nodes in an isolated subgraph (and all adjacent arcs) can thus be discarded. We then return to the reduction phase, since this could enable additional reductions in the graph.

Let us now assume that a graph with at least three nodes and at least one eligible arc remains. We also assume that no valid cycle has been found, no more reduction of the graph is possible and that the heuristic cycle search fails to find a cycle.

Suppose that we choose an eligible arc (i, j), say a forward arc not in $A_{l'}$. A cycle including this arc consists of arc (i, j) and a path from node j to node i. Therefore we search for a path from j to i (or a path from i to j if (i, j) is a backward arc). We set cost zero on all eligible arcs and one on all others. Arcs labeled with both F and B are duplicated, one in each direction.

Then we find the shortest path from node j to i with a standard shortest path method, for example Dijkstra's method, which has the complexity $O(|N|^2)$. The result is either a path from j to i, or a proof (a cut separating j from i) that there exists none. If we have found a path, a cycle is formed by adding arc (i, j), and we have succeeded in finding a valid cycle. If there is no path, we know that there exists no cycle including arc (i, j). Then this arc can be removed from the graph, and we return to the reduction phase, which may yield new results in the absence of arc (i, j).

Actually, using Dijkstra's method, we will label all nodes that are reachable from node j, and the cut will indicate a set of nodes, D, that has no leaving arc. If there is no eligible arc in the subgraph spanned by D, we have found an isolated subgraph which can be eliminated.

This way we will either find a valid cycle or eliminate the whole graph. Concerning the complexity, at least one arc will be removed in each main iteration, after which the reduction phase is redone. A crude estimation of the complexity of the method is $O(m^2|A||N|^3)$, which is $O(|A||N|^5)$ if the number of SP-graphs equals the number of nodes, and it is certainly not more than $O(|N|^7)$. This can probably be decreased, but our conclusion is that the method is polynomial. Furthermore, this complexity has little to do with the practical performance of the method.

Let us summarize the Valid Cycle (VC) method in a more algorithmic fashion.

1. **Choice of SP-graphs:** If all pairs of SP-graphs have been compared, go to 12. Otherwise choose two SP-graphs l' and l'' not previously compared.

2. **Labeling phase:** Label each arc in $A_{l'}$ with B and each arc in $A_{l''}$ with F. Arcs labeled with only B or only F are marked eligible.

3. **Reduction phase:** Repeat until no more changes:
   Remove nodes with only one adjacent arc.
   Remove nodes with no entering (leaving) arcs.
   Remove an arc label if the other label gives the only entering (leaving) arc.
   Remove all isolated components with less then three nodes or arcs.

4. **Elimination check:** If all arcs are eliminated, go to 11 .

5. **Eligible arc:** Search the remaining graph for an eligible arc. If no eligible arc exist, go to 11. Otherwise, let (i, j) be the eligible arc found, and L its label.

6. **Heuristic cycle search:** Find a cycle by heuristic: Start with arc (i, j) (in the proper direction) and traverse nodes, using adjacent arcs. Never use the same arc twice. Stop when a node is visited a second time: A cycle is found.

7. **Evaluation of cycle:** If the found cycle contains an eligible arc, go to 13. If the found cycle has total in-degree or out-degree equal to zero, go to 10.

8. **Shortest path cycle search:** Set arc cost equal to zero for eligible arcs and equal to one for the other arcs. Find shortest path from node j to node i if L is F, or from node i to node j if L is B. If a path exists, add arc (i, j) to form a cycle, and go to 13.

9. **No path:** Remove arc (i, j). If the reachable subgraph contains some eligible arc, go to 3.

10. **Isolated subgraph:** An isolated subgraph is found. Eliminate all nodes in the subgraph and all adjacent arcs. Go to 3.

11. **Graph eliminated:** No valid cycle found. Go to 1.

12. **No valid cycle found:** No valid cycle exists. Terminate the method. (Try to find compatible weights.)

13. **Valid cycle found:** A valid cycle is found. Terminate the method. No compatible weights exist.

Comments: Feasible directions of the arcs are given by the labels. When a node is removed in the reduction phase, all adjacent arcs are also removed. When searching for cycles, the direction is given by the label of the eligible starting arc. A cycle found by the shortest path method always contains an eligible arc, so this does not need to be checked.

For the special case when all SP-graphs are trees, theorem 3 tells us that any feasible cycle is also improving, and thus valid. A feasible cycle is always found in step 6, and we know that it is valid, so we will go directly to step 13. In this case, steps 7, 8, 9 and 10 will never be used, and can be removed from the algorithm.

Theorem 4 *After a finite number of steps, algorithm VC will terminate, either with a valid cycle, or with the whole graph eliminated, in which case there exists no valid cycle.*

The algorithm VC can be used to check if a number of SP-graphs agree. See section 6 for a discussion about the possibilities of changing SP-graphs to make them agree. It can also be used in different iterative procedures for determining which SP-graphs to use, out of a larger number. One might also consider to use it within an advanced *Constraint Programming* method, where algorithm VC is an implemented constraint.

6 Modifications of SP-graphs

If no compatible set of weights exist for a certain set of SP-graphs, these SP-graphs can not be realized in an IP network using OSPF. In this context there is something "wrong" with this set of SP-graphs.

If our task is to determine the values of the weights, and we are forced to use the weights found, we will get traffic patterns that are different from what is desired. However, as P1 is infeasible, solving it will not yield any useful information about how to set the weights.

P1 could be made feasible by changing some of the indata. The simplest possibility is to remove some SP-graph, which means that some paths are no longer considered to be desired. In such a case, our method is directly useful, since it indicates two conflicting SP-graphs, l' and l''. Removing one of these will remove that particular conflict. Obviously this may have to be repeated, since our method stopped when it found the first conflicting pair of SP-graphs.

Another possibility is to modify an SP-graph. Again it is useful that our method indicates which SP-graphs to consider. We even know which set of arcs in the SP-graphs to consider. In such a situation, we get a cycle $C = F \cup B$ and two indices l' and l'', such that $B \subseteq A_{l'}$ and $F \subseteq A_{l''}$ (since it is feasible). We would also like to have $B \subseteq A_{l''}$ and $F \subseteq A_{l'}$, but there is at least one arc not fulfilling this, since the cycle is improving. The arcs making it improving are given by the sets $\hat{F} = \{(i,j) : (i,j) \in F, (i,j) \notin A_{l'}\}$ and $\hat{B} = \{(i,j) : (i,j) \in B, (i,j) \notin A_{l''}\}$.

In order to remove a valid cycle, we can either make it non-improving or make it infeasible (or both). The following actions are possible to take.

1. Add all arcs in $\hat{F}$ to $A_{l'}$ and and all arcs in $\hat{B}$ to $A_{l''}$. This makes the cycle non-improving.

2. Remove one arc in $B \cap A_{l'}$ from $A_{l'}$ or one arc in $F \cap A_{l''}$ from $A_{l''}$ in such way that the SP-graph remains connected. This makes the cycle infeasible.

3. Replace arc(s) in $A_{l'}$ or $A_{l''}$ to make the cycle infeasible and/or non-improving.

Unfortunately, these changes might create new conflicts between the SP-graphs. Adding arcs might make a previously infeasible cycle feasible, and if it is improving, it will become valid. Removing arcs might make a previously non-improving cycle improving, and if it is feasible, it will be valid. The development of a better method for changing SP-graphs is a topic for future research.

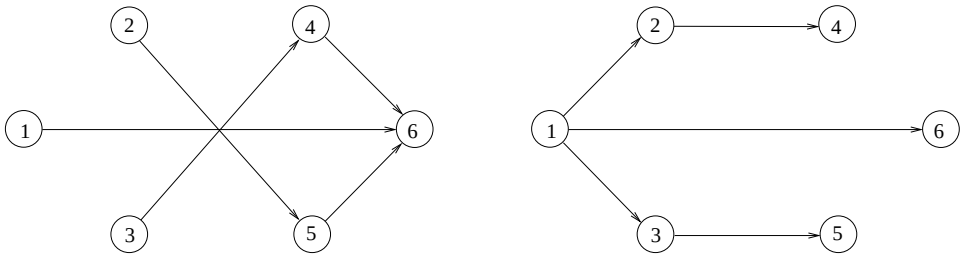


Figure 2: The in-tree A_1 and out-tree A_2.

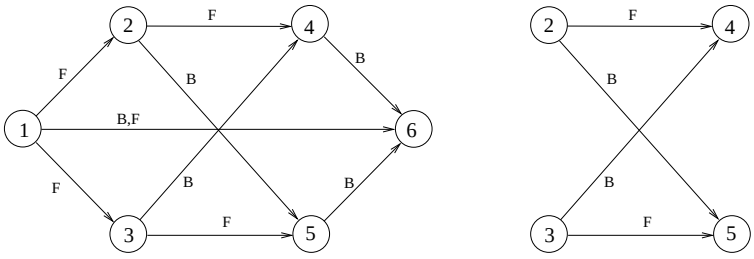


Figure 3: The graph after the labeling phase (left) and the graph after the reduction phase (right).

7 A small example

Let us now study an example with two SP-graphs (based on an example given in [7]). SP-graph A_1 is an in-tree entering node 6, while SP-graph A_2 is an out-tree leaving node 1, as shown in figure 2. This pair of SP-graphs is suboptimal, since the only node pair connected in both A_1 and A_2 is connected by an identical path (node pair (1,6)).

We label all arcs in A_1 with B and all arcs in A_2 with F. The left graph in figure 3 shows the situation after the labeling phase.

In the graph reduction phase, arcs (1,2) and (1,3) are labeled F so if it should be possible to enter node 1, arc (1,6) must be labeled B. Therefore we remove label F from arc (1,6). However, now node 6 can not be entered, so node 6 and all adjacent arcs are discarded. Since arc (1,6) was removed, it is not possible to enter node 1, so it is discarded, together with all adjacent arcs. In the right part of figure 3, we show the graph remaining after the reduction phase.

All arcs are now eligible, since both arcs in F lie outside A_1 and both arcs in B lie outside A_2. The heuristic cycle search will find the cycle 2 - 4 - 3 - 5 - 2 and we conclude that there does not exist any compatible weights for this pair of SP-graphs.

If we wish to modify the SP-graphs so that compatible weights exist, we could add arcs (2,4) and (3,5) to A_1 and (2,5) and (3,4) to A_2. There are now two paths in A_1 between nodes 2 and 6 (and nodes 3 and 6), so this introduces splitting of the traffic. The same happens in A_2 between nodes 1 and 4, and between 1 and 5.

We might instead consider removing an arc from A_1 or A_2. However, as A_1 and A_2 are trees, this is not possible since the corresponding SP-graph will fall apart. It is possible to replace an arc in one SP-graph, for example replace (2,5) by (2,4) in A_1. This change will remove the valid cycle. Furthermore no new conflicts will appear, and there exists compatible weights after the change.

Table 1: Computational results.

$\|N\|$	P	N_V	N_W	N_O	T_T	T_I
10	41	18	22	1	16	0.4
15	42	35	7	0	34	0.8
20	40	33	7	0	97	2.4
30	40	33	7	0	536	13.4

8 Implementation and computational tests

The VC-algorithm has been implemented in Tcl/Tk within the framework of the graphical package VINEOPT (Visual Network Optimization) (www.vineopt.com). Tcl/Tk is a scripting language, and the code has been translated to C with the package MKTCLAPP. Nevertheless this implementation is probably much slower than a proper implementation in C.

A number of test problems have been generated in the following manner. We start with four networks with 10, 15, 20 and 30 nodes and generate weights such that splitting is probable. Shortest path trees to each node are calculated, and SP-graphs are constructed by including all arcs with reduced cost equal to zero. This means that the number of SP-graphs is equal to the number of nodes, and that all SP-graphs are in-graphs spanning all nodes.

The resulting SP-graphs obviously have compatible weights. A goal of the problem generation is that it should not be known in advance if compatible weights exist, so some modifications are made. The first modification is to include one arc with reduced cost equal to one in a randomly chosen SP-graph. The second modification is to add an arc to a randomly chosen SP-graph, such that the depth of the ending node of the arc is greater than the depth of the starting node. It can be shown that neither of these modifications lead to directed cycles in the SP-graphs.

One solution approach is to first try to find compatible weights by solving P1 with an LP-code, and if it fails to find a feasible solution, analyze the situation further with the VC-algorithm. Another approach is to first run the VC-algorithm, which will either prove that no compatible weights exist, or indicate that compatible weights might exist. In the latter case, we try to find the weights by solving P1.

The first approach is probably more efficient, since all pairs of SP-graphs must be considered by the VC-algorithm. However, since the topic of this paper is to analyze a set of SP-graphs, we have chosen to use the second approach. The largest part of the solution time is therefore spent on comparing the SP-graphs, since P1 is only solved when no valid cycle is found. We use the code LPSOLVE for solving P1.

In table 1 the computational results are summarized. P denotes the number of instances in the group, N_V is the number of instances with valid cycles, N_W is the number of instances with compatible weights, N_O is the number of instances without valid cycles and compatible weights (which means that the unbounded solution of P3 is of a more complicated type). T_T is the total time for all of the problems, and T_I is the average time for each instance, both in seconds. The computer used is a 2.4 GHz PC running Linux.

Table 1 shows that the solutions times are reasonable. Another important result is that only for *one* of the 163 different instances solved, compatible weights do not exist even though no valid cycle is found. This indicates that valid cycles seem to capture most cases where compatible weights do not exist. We can therefore draw the conclusion that valid

cycles give a practical and useful characterization of instances for which no compatible weights exist.

9 Conclusions

We have presented a new and useful way of finding instances of shortest paths graphs that together prohibits the existence of compatible weights for IP networks using OSPF. The characterization is based on so called valid cycles. A polynomial method for finding valid cycles is proposed. Computational results confirm that the solution method does not take exceedingly long time, and that it seems to capture most cases where compatible weights do not exist.

Future research will consist of including this method into a method for finding the optimal design of an OSPF network. We will also investigate and develop practical solution methods for the more complicated cases where neither compatible weights nor valid cycles exist.

Acknowledgment: This work has partly been financed by the Swedish Research Council.

Bibliography

[1] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B., *Network Flows. Theory, Algorithms and Applications*, Prentice Hall 1993.

[2] Ben-Ameur, W. and Gourdin, E., "Internet routing and related topology issues", *SIAM Journal on Discrete Mathematics* 17 (2003) 18–49.

[3] Broström, P. and Holmberg, K., "Determining the non-existence of a compatible OSPF metric", Research Report LiTH-MAT-R-2004-06, Department of Mathematics, Linköping Institute of Technology, Sweden 2004.

[4] Farago, A., Szentesi, A., and Szviatovszki, B., "Allocation of administrative weights in PNNI", in: *Proceedings of the Networks'98, Sorrento*, pages 621–625, 1998.

[5] Farago, A., Szentesi, A., and Szviatovszki, B., "Inverse optimization in high-speed networks", *Discrete Applied Mathematics* 129 (2003) 83–98.

[6] Fortz, B. and Thorup, M., "Internet traffic engineering by optimizing OSPF weights", in: *Proceedings of IEEE INFOCOM '00* volume 2, pages 519–528, 2000.

[7] Gourdin, E., "Optimizing internet networks", *ORMS Today* 28/2 (2001) 46–49.

[8] Holmberg, K. and Yuan, D., "Optimization of Internet Protocol network design and routing", *Networks* 43(1) (2004) 39–53.

[9] Pioro, M., Szentesi, A., Harmantos, J., Jüttner, A., Gajowniczek, P., and Kozdrowski, S., "On open shortest path first related network optimization problems", *Performance Evaluation* 48 (2002) 201–223.

Cutting Plane Methods in Decision Analysis

Xiaosong Ding[1] and **Faiz Al-Khayyal**

xiaosong.ding@gmail.com, Department of Information Technology and Media, Mid-Sweden University, SE-851 70, Sundsvall, Sweden.

faiz.alkhayyal@isye.gatech.edu, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA.

Abstract

Several computational decision analysis approaches have been developed over a number of years for solving decision problems when vague and numerically imprecise information prevails. However, the evaluation phases in the DELTA method and similar methods often give rise to special bilinear programming problems, which are time-consuming to solve in an interactive environment with general nonlinear programming solvers. This paper proposes a linear programming based global optimization algorithm that combines the cutting plane method together with the lower bound information for solving this type of problems. The central theme is to identify the global optimum as early as possible in order to save additional computational efforts.

[1] Corresponding author

1 Introduction

With the rapid development of graphical user interfaces, it is possible to bring the use of sophisticated computational techniques for decision analysis to a broader group of users, and many decision analytic tools have emerged. However, most of them consist of some straightforward set of rules applied to precise numerical estimates of probabilities and values no matter how unsure a decision maker is of his or her estimates. The requirement to provide numerically precise information in such models has often been considered unrealistic in real-life decision situations. Besides, sensitivity analysis is often not easy to carry out in more than a few dimensions at a time because of precise figures. When a decision maker is faced with a decision problem that could not be directly judged by his or her empirical experience, or according to available historical data, a module allowing imprecision is obviously of great value.

A number of techniques allowing imprecise statements have been suggested, but they are concentrated more on representation and less on evaluation. In spite of several years of intense activities, only a few decision analytic tools, for example, ARIADNE, *Decide*IT and Winpre, can evaluate imprecise estimates. Among these tools, the DELTA method for decision analysis, described in [4, 5, 6, 7, 11], is an approach towards analyzing decision problems containing imprecise information, represented by intervals and relations. It has been implemented in the Decision Analysis System (DAS) *Decide*IT [8], and has been used in various real-life contexts; e.g., [12]. Due to the introduction of interval and qualitative estimates, the relaxation of classical decision theory in this respect gives rise to special Bilinear Programming (BLP) problems, whose study is a sub-field of Nonlinear Programming (NLP).

In Fig. 1 below, a decision tree is presented,

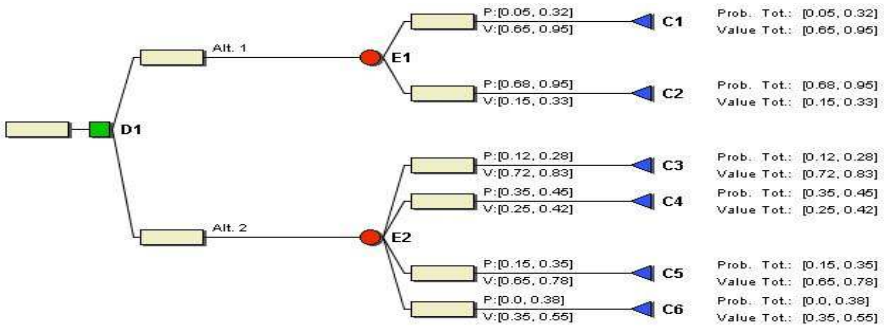


Figure 1: A Decision Tree

where $D1$ is a decision node, $E1$ and $E2$ are probability nodes, representing indeterminism, with associated probability distributions. The leaves are consequence

nodes with convex sets of associated value or utility functions. In DELTA, a *decision frame* represents a decision problem of this type. The idea behind such a frame is to collect all information necessary for the model in one structure. This structure is then filled in with user statements represented as linear inequalities. User statements can be range constraints, core intervals, or comparative statements. In a decision frame, a consequence c_i is denoted by a variable v_i and the user statements can be of the following forms for the numbers a_1, a_2, b_1, b_2, d_1, and d_2.

- Range: v_i is definitely between a_1 and a_2;
- Core interval: v_i is likely to fall between b_1 and b_2; and
- Comparison: v_i is larger than another variable, v_j, by an amount between d_1 and d_2.

All value statements are translated and collected together in a *value base* (*V-base*). On the other side, with the usual normalization constraints $\sum_{i\in I} p_i = 1$ and $\sum_{j\in J} p_j = 1$, where I and J are index sets labelling the consequences of two alternatives, all probability statements in a decision problem are translated into a *probability base* (*P-base*). The structure $\prec P, V \succ$ is referred to as a *decision frame*.

Given a *decision frame* $\prec P, V \succ$, the primary evaluation rules in DELTA are based on pair-wise comparisons using a generalization from the principle of maximizing the expected utility. A typical issue in this context is to maximize an expression, such as $\max(\sum_{i\in I} p_i v_i - \sum_{j\in J} p_j v_j)$, which is subject to a constraint set defined by a decision frame. Similar evaluation rules apply in other analysis methods.

More generally, comparative decision rules in computational decision analysis are variations of the following form:

$$\tfrac{1}{2}[\min(\textstyle\sum_{i\in I} p_i v_i - \sum_{j\in J} p_j v_j) + \max(\sum_{i\in I} p_i v_i - \sum_{j\in J} p_j v_j)] \tag{1}$$

In a typical decision situation, imprecise estimates occur in both *P-* and *V-bases*, which results in a special BLP problem. It should be noted that in (1), the corresponding maximization problem is readily solved by minimizing the negation of a minimization problem. Therefore, without losing any generality, throughout the rest of this paper, the focus will be centered on developing a rapid BLP algorithm for solving:

$$\begin{array}{ll} \min & f(p,v) = \sum_{i\in I} p_i v_i - \sum_{j\in J} p_j v_j, \\ \text{s.t.} & \left[\begin{array}{c} L_P \\ L_V \end{array}\right] \le \left[\begin{array}{cc} C_P & 0 \\ 0 & C_V \end{array}\right] \cdot \left[\begin{array}{c} P \\ V \end{array}\right] \le \left[\begin{array}{c} U_P \\ U_V \end{array}\right] \end{array} \tag{2}$$

where L_P, C_P and U_P represent the lower bounds, constraint coefficients and upper bounds in the *P-base*, respectively; $P^t = (p_I^t, p_J^t)$ represents the variables in the *P-base*; and by analogy, these definitions also exist in the *V-base*.

The next section will describe the optimization background employed in our procedure, which is followed by developing a BLP algorithm that combines a cutting plane method in a local optimization phase with a lower bounding method in a global optimization phase. Then a numerical example is solved to illustrate the

elements of the BLP algorithm. Computational results on more than four-hundred randomly generated decision analysis problems indicate that the approach is very effective for solving practical sized decision analysis problems in real time on a laptop architecture computer. Two possible directions for future research on our approach are suggested in the final section.

2 Optimization

Consider the standard disjoint BLP problem:

$$\begin{array}{ll} \min & f(x,y) = c^t x + d^t y + x^t C y, \\ \text{s.t.} & x \in X_0 = \{x \in R^m : A_1 x \leq b_1, x \geq 0\}, \\ & y \in Y_0 = \{y \in R^n : A_2 y \leq b_2, x \geq 0\} \end{array} \tag{3}$$

where $c \in R^m$ and $d \in R^n$ are linear parts for x and y, respectively, $C \in R^{m \times n}$, and X_0 and Y_0 are bounded polyhedral sets. In terms of (2), both c and d are zero vectors, and C is always an indefinite square matrix with only $+1$ or -1 diagonal elements. For example, in Fig. 1:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

The disjoint BLP (3) is one type of general Quadratic Programming (QP) problems with a symmetric indefinite quadratic form matrix. The special cases (2) that arise in computational decision analysis have the added property that the bilinear form matrix C is indefinite. In both cases, the problem is non-convex and global optimization strategies are required to find the absolute minimum objective value, which is called the *global minimum*, and its corresponding solution point, which is called the *global minimizer*. It should be noted that we distinguish between the minimal objective function value and the corresponding point at which it is achieved as *minimum* and *minimizer*, respectively. A general framework for many global optimization strategies is summarized in [18] as:

> "Actually all methods for global optimization consist of two phases: a global phase, aimed at thorough exploration of the feasible region or subsets of the feasible region where it is known the global optimum will be found, and a local phase aimed at locally improving the approximation to some local optima. Often these two phases are blended into the same algorithm, which automatically switches between exploration and refinement."

The procedure presented herein captures the spirit of this general framework by proposing a refinement to an existing algorithm for the local phase blended with a formulation for the global phase to produce a global optimization algorithm that finds either an exact global minimizer or an epsilon-global minimizer with a specified tolerance.

An important property of (3) to observe is that even though $f(x, y)$ can be shown to be not quasi-concave, an optimal solution (x^*, y^*) exists at an extreme point of $X_0 \times Y_0$, [2]; i.e., x^* is an extreme point of X_0 and y^* is an extreme point of Y_0. However, this property is lost in the jointly constrained case such as:

$$\begin{array}{ll} \min & f(x,y) = c^t x + d^t y + x^t C y, \\ \text{s.t.} & x, y \in \{x \in r^m, y \in r^n : A_1 x + A_2 y \leq b, x \geq 0, y \geq 0\} \end{array} \tag{4}$$

To solve a jointly constrained BLP problem with a non-extremal boundary point optimum poses the greatest computational difficulty. However, for those BLP problems exhibiting extreme point optimal solutions, it is relatively easy to solve. Some computational results are reported in [20, 21].

The other important property of (3) is that any cuts involving variables associated with both X_0 and Y_0 sets will destroy their special structures. Problems do exist where one of the sets has special structure that can be exploited by efficient algorithms which can be used to solve sub-problems in the solution procedure, [23]. Accordingly, we prefer developing linear cuts within only one polyhedron.

3 Local Optimization

The local optimization phase aims at locating a local optimum. Any local optimization technique for finding *Karush-Kuhn-Tucker* (KKT) solutions of quadratic programs, such as Wolfe's simplex method or an interior point method, can accomplish this task. Nevertheless, the structure of the disjoint BLP problem (3) itself suggests a Linear Programming (LP) based vertex following algorithm, which is very convenient and efficient, [16]. The approach consists in starting with an arbitrary fixed $x \in X_0$, and solving the related LP problem with y as the unknown. The solution, y, is then used to solve another LP problem with x as the unknown. This in turn yields a new solution for x. The procedure is repeated until a pair of values $(\overline{x}, \overline{y})$ is found that solves both LP problems. It has been proved that the resulting solution is a KKT point.

DEFINITION 1: Consider $P : \min f(x)$ subject to $x \in S$, where S is a compact polyhedral set and f is non-convex. A *local star minimizer* of P is defined as a point $\overline{x}$ such that $f(\overline{x}) \leq f(x)$ for each $x \in N(\overline{x})$, where $N(\overline{x})$ denotes the adjacent extreme points to $\overline{x}$.

Extending the definition of $N(\overline{x})$ into the disjoint BLP (3), an extreme point is

adjacent to $(\overline{x}, \overline{y})$ if and only if it is of the form $(x^i, \overline{y})$ or $(\overline{x}, y^i)$ where $x^i \in N(\overline{x})$ and $y^i \in N(\overline{y})$.

DEFINITION 2: An extreme point is called a *pseudo-global minimizer* if $f(\overline{x}, \overline{y}) \leq f(x, y)$ for each $x \in B_\delta(\overline{x}) \cap X_0$ and for each $y \in Y_0$, where B_δ is a δ neighborhood around $\overline{x}$.

Intuitively, a pseudo-global minimizer is an extreme point that satisfies the KKT conditions, has no descent directions within its neighborhood, and acts as a local minimizer in x-space and a global minimizer in y-space. In order to obtain a pseudo-global minimizer, we closely follow the algorithm described in [26].

ALGORITHM 1:

1. Find a feasible extreme point x^1 in X_0.
2. [a] Solve: $\min\{f(x^1, y) | y \in Y_0\}$, to yield an optimal y^1.
   [b] Solve: $\min\{f(x, y^1) | x \in X_0\}$, to yield an optimal x^2.
   Repeat until the procedure converges to a local star minimizer $(\overline{x}, \overline{y})$.
3. Let $x^1, \ldots, x^m$ be the adjacent extreme points of $\overline{x}$.
   Solve: $\min\{f(x^i, y) | y \in Y_0\}$, to yield solutions $y^1, \ldots, y^m$.
4. If $f(\overline{x}, \overline{y}) \leq f(x^i, y^i)$ for all i, terminate with $(\overline{x}, \overline{y})$ as a pseudo-global minimizer.
5. Choose one $f(x^r, y^r) \leq f(\overline{x}, \overline{y})$ and go back to 2[b] with $y^1 = y^r$.

The performance to locate a KKT point in the DELTA method has been reported in [10]. Based on computational observations, a KKT point is found within four iterations, on average. However, checking its adjacent extreme points is relatively time-consuming, especially when we have to return to step 2[b] from step 5.

4 Global Optimization

Given a pseudo-global optimizer, a linear cut needs to be generated within only one polyhedron. The cutting plane techniques for bilinear programs were inspired by similar methods for concave problems, [19, 25]. One of the first such procedures was proposed in [16] to delete local vertex solutions by using Ritter's cut [19]. Another cutting plane approach was developed in [13] by using Tuy's cut [25]. The latter used LP duality theory to reformulate the BLP problem as an equivalent concave minimization problem with an implicitly defined objective function. The polar cuts of [3] were applied in [26] to BLP, where it was proved that the polar cuts uniformly dominate other similar cuts. This approach was further strengthened in [22] by employing negative edge extension polar cuts and disjunctive face cuts, whereupon finite convergence to an exact solution could be guaranteed. In [15], it has been pointed out that [22] might be the most efficient approach for handling

BLP problems. Accordingly, in this paper, we employ polar cuts to cut off local vertex solutions.

Let $\overline{x}$ be an extreme point of X_0 and let $p_j, j \in J$, be the m nonbasic variables at $\overline{x}$, where J is the index set for the nonbasic variables. Denoting by $\overline{e}^j$ the columns of the simplex tableau in extended form, the m-vector x can be written as:

$$x = \overline{x} - \textstyle\sum_{j \in J} \overline{e}^j p_j.$$

Barring the degenerate case, X_0 has precisely m distinct edges incident to $\overline{x}$ and each half line

$$\xi^j = \{x : x = \overline{x} - \overline{e}^j \lambda_j, \lambda_j \geq 0\}, j \in J \tag{5}$$

contains exactly one such edge, [3].

DEFINITION 3: The *generalized reverse polar* of Y_0 for a given scalar α is given by $Y_0(\alpha) = \{x : f(x, y) \geq \alpha\}$ for all $y \in Y_0$.

Following [22, 26], let $(\overline{x}, \overline{y})$ be a pseudo-global minimizer, let the rays ξ^j be defined as in (5), let α be the current best objective value of $f(x, y)$, and let $\overline{\lambda}_j$ be defined by:

$$\overline{\lambda}_j = \begin{cases} \max\{\lambda_j : f(\overline{x} - e^j \lambda_j, y) \geq \alpha \text{ for all } y \in Y_0\} \text{ if } \xi^j \not\subset Y_0(\alpha), \\ -\max\{\lambda_j : f(\overline{x} + e^j \lambda_j, y) \geq \alpha \text{ for some } y \in Y_0\} \text{ if } \xi^j \subset Y_0(\alpha) \end{cases} \tag{6}$$

Then the inequality

$$\textstyle\sum_{j \in J} p_j / \overline{\lambda}_j \geq 1 \tag{7}$$

is a valid cutting plane. The second line in (6) is referred to as the negative extension polar cuts. Inequality (7) is a valid cut in the sense that firstly, it does not contain the current pseudo-global optimum; and secondly, it contains all the candidates $x \in X_0$ for which $\min\{f(x, y) | y \in Y_0\} < \alpha$.

The cutting plane method searches for the global optimum by exhausting all possibilities within one of the two bounded polyhedra. Although ALGORITHM 1 always generates a feasible solution, we have no way of knowing if we have found the global solution until we have cut off all of the pseudo-global optima. Consequently, the key idea is to obtain some lower bound information concerning the global solution. Then at least we can tell how close the current best solution is to the global optimality before an exhaustive search. We employ the convex and concave envelopes of $x_i y_i$ developed in [1, 2] to obtain such a lower bound. Consider the inner product $x^t y$ over the compact hyper-rectangle $\Omega = \{(x, y) : l \leq x \leq L, m \leq y \leq M\}$. Define $\Omega_i = \{(x_i, y_i) : l_i \leq x_i \leq L_i, m_i \leq y_i \leq M_i\}$ so that $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$. The convex and concave envelopes of $x_i y_i$ over Ω_i are defined as:

$$\begin{aligned} Vex_{\Omega_i}[x_i y_i] &= \max\{m_i x_i + l_i y_i - l_i m_i, M_i x_i + L_i y_i - L_i M_i\}, \\ Cav_{\Omega_i}[x_i y_i] &= \min\{M_i x_i + l_i y_i - l_i M_i, m_i x_i + L_i y_i - L_i m_i\} \end{aligned} \tag{8}$$

Accordingly, in (2), we can calculate the convex envelopes for $C_{ii} = 1$ and concave envelopes for $C_{ii} = -1$. Then we can say:

$$\begin{array}{ll} \min & f(p,v) = \sum_{i\in\{i:C_{ii}=1\}} Vex_{\Omega_i}[p_i v_i] - \sum_{i\in\{i:C_{ii}=-1\}} Cav_{\Omega_i}[p_i v_i], \\ \text{s.t.} & \left[\begin{array}{c} L_P \\ L_V \end{array}\right] \leq \left[\begin{array}{cc} C_P & 0 \\ 0 & C_V \end{array}\right] \cdot \left[\begin{array}{c} P \\ V \end{array}\right] \leq \left[\begin{array}{c} U_P \\ U_V \end{array}\right] \end{array} \tag{9}$$

is an underestimating problem for (2), whose solution yields a lower bound on the global minimum of our bilinear decision problem. In (9), the rectangles Ω_i define the bounds on p_i and v_i which are both readily available.

Since solving (9) yields a lower bound on the optimal objective value of (2), if the algorithm cuts off a pseudo-global optimizer with a polar cut and proceeds to search for another one in the smaller set, then we can solve the underestimating problem with the convex and concave envelopes computed over the smaller region to obtain a tighter bound on all global solutions in the reduced feasible set. If the algorithm cuts off the global solution and the objective of the underestimating problem is higher than the current best objective value, then we can stop and use the current best solution as the global solution. If there are many global optimal solution points, the objective of the underestimating problem will be smaller than the global value until all global solution points have been cut off.

If the feasible region has not been exhausted and the underestimating problem is still giving optimal values lower than the current best solution, then it is always possible to stop the search procedure early with a known feasible point and a lower bound on the global optimum. In that case, an error bound will be available to show how far we are away from global optimality in the worst scenario.

Denote by X_0 the original feasible region or its subset obtained after the introduction of generated polar cuts. The global optimization algorithm for (2) can be summarized as follows:

ALGORITHM 2:

1. Let the best objective value, *obj*, be a large positive number, and let an epsilon tolerance, ϵ, be a prescribed small number.
2. Calculate the lower bound, *bound*, for X_0^i by using (9).
3. If $bound > obj$ or $|bound - obj| \leq \epsilon$, or the unexplored feasible region X_0^i at stage i is empty, terminate with *obj* as the global minimum.
4. Find a pseudo-global minimizer by using ALGORITHM 1, and update *obj* if necessary.
5. If $|bound - obj| \leq \epsilon$, terminate with *obj* as the global minimum.
6. Solve m LP problems by using (6) to obtain $\overline{\lambda}_j$, and generate the polar cut by using (7), and introduce it into X_0^i.
7. increase i to $i + 1$, go back to 2.

We do not employ the extreme face finding routine and disjunctive face cuts as in [22] because they are relatively expensive to calculate. Instead, we take advantage of ALGORITHM 1 to locate a pseudo-global optimizer. As pointed out in the last section, ALGORITHM 1 is also time-consuming if it proves necessary to frequently locate a new local star minimizer. Therefore, it is difficult to determine which procedure is more efficient from a computational viewpoint. ALGORITHM 2 simply adds the lower bound computation in order to more quickly identify when a global optimizer has been found.

5 Numerical Example

In this section, a numerical example will be used to illustrate ALGORITHM 2. The data for this experiment is randomly generated by Matlab to simulate a real-life decision situation, [9].

Suppose now we have a decision situation consisting of two alternatives with six consequences in each alternative. Correspondingly, $P = (p_{11}, \ldots, p_{16}, p_{21}, \ldots, p_{26})^t$ and $V = (v_{11}, \ldots, v_{16}, v_{21}, \ldots, v_{26})^t$. The matrices C_P and C_V are given below:

$$C_P = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$C_V = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

In C_P, each entry represents the coefficient of each variable. For example, the first and second lines are normalization requirements, $\sum_{i \in I} p_i$ and $\sum_{j \in J} p_j$, with respect to each alternative, whereas the third line means $p_{21} - p_{25}$, and etc. The contents in C_V are explained analogously. In practice, the value base does not contain compound value statements since they lack semantic content.

The bounds L_P, L_V, U_P and U_V are listed in Table 1. In addition, each variable in the *P-base* is restricted within the interval $[0, 1]$, and each variable in the *V-base* is in:

$$\begin{bmatrix} 0.151 \\ 0.210 \\ 0.080 \\ 0.277 \\ 0.740 \\ 0.340 \end{bmatrix} \leq \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{14} \\ v_{15} \\ v_{16} \end{bmatrix} \leq \begin{bmatrix} 0.866 \\ 0.592 \\ 0.174 \\ 0.541 \\ 0.791 \\ 0.593 \end{bmatrix}, \quad \begin{bmatrix} 0.083 \\ 0.018 \\ 0.780 \\ 0.156 \\ 0.020 \\ 0.057 \end{bmatrix} \leq \begin{bmatrix} v_{21} \\ v_{22} \\ v_{23} \\ v_{24} \\ v_{25} \\ v_{26} \end{bmatrix} \leq \begin{bmatrix} 0.480 \\ 0.303 \\ 0.848 \\ 0.354 \\ 0.152 \\ 0.637 \end{bmatrix}.$$

C_P :	Rows	L_P	U_P	C_V :	Rows	L_V	U_V
	1	1.000	1.000		1	-1.692	1.692
	2	1.000	1.000		2	-0.576	0.061
	3	-0.437	0.182		3	-0.141	0.510
	4	-0.296	0.394		4	-1.182	-0.302
	5	-0.131	0.313		5	-0.576	0.297
	6	-0.543	0.376		6	-0.523	0.456
	7	0.773	1.692				

Table 1: Data

If we simply use the cutting plane method, the feasible region of the *P-base* will be exhausted in four cuts. Nevertheless, the second lower bound information in Table 2 is enough to guarantee the global optimum; i.e., the stopping rule, $bound > obj$, is satisfied in ALGORITHM 2, step 3. Therefore, ALGORITHM 2 terminates in one iteration; thus, saving the additional computational effort of performing three more iterations.

Iteration	Objective Value	Lower Bound
1	-0.60742032702968	-0.61148735388339
2	-0.54069572061830	-0.54476274747200
3	-0.51501142176621	-0.51907844861991
4	-0.46983913868983	-0.47390616554353

Table 2: An Numerical Example

6 Computational Experience

We launched a number of simulated instances to test ALGORITHM 2. The experiment is performed on a personal computer with Windows 2000, Matlab 6.5 & Tomlab [24], Pentium-III 1000 MHz CPU and 512MB memory. The commercial LP

solver is SQOPT [14] from Systems Optimization Laboratory, Stanford University. In total, we tested 400 instances consisting of 10 groups, each of them including 40 data sets with the consequences from 11 to 50. If an instance has N consequences, the corresponding BLP problem contains 4N variables and around 2N constraints. Consequently, this experiment is trying to solve disjoint BLP problems sized up to 200 variables. Detailed numerical results are shown in Table 3.

C	P_V	V_V	P_C	V_C	$Time$	C	P_V	V_V	P_C	V_C	$Time$
11	22	22	11	10	2.0901	31	62	62	31	30	7.7427
12	24	24	14	12	2.5595	32	64	64	34	32	13.3900
13	26	26	14	12	3.8995	33	66	66	34	32	8.8322
14	28	28	15	14	2.0418	34	68	68	35	34	11.1826
15	30	30	15	14	2.3370	35	70	70	35	34	12.1919
16	32	32	18	16	5.4044	36	72	72	38	36	12.6346
17	34	34	18	16	3.8734	37	74	74	38	36	15.7232
18	36	36	19	18	4.6272	38	76	76	39	38	12.4650
19	38	38	19	18	4.0508	39	78	78	39	38	18.9598
20	40	40	22	20	7.9871	40	80	80	42	40	20.8625
21	42	42	22	20	5.0642	41	82	82	42	40	16.7570
22	44	44	23	22	5.9443	42	84	84	43	42	12.0811
23	46	46	23	22	7.1294	43	86	86	43	42	19.5244
24	48	48	26	24	5.5255	44	88	88	46	44	17.6758
25	50	50	26	24	5.9522	45	90	90	46	44	25.3698
26	52	52	27	26	6.8825	46	92	92	47	46	22.2200
27	54	54	27	26	8.1232	47	94	94	47	46	30.8439
28	56	56	30	28	10.9514	48	96	96	50	48	19.9436
29	58	58	30	28	7.5786	49	98	98	50	48	20.9942
30	60	60	31	30	9.8275	50	100	100	51	50	19.3494

- C represents the number of consequences;
- $P_V = V_V$ represent the number of variables in *P-base* and *V-base*;
- $P_C = V_C$ represent the number of constraints in *P-base* and *V-base*;
- $Time$ is the average CPU time in seconds.

Table 3: Detailed Results

As for the worst case, the global optimum might not be found until we cut off all pseudo-global optimizers, and the lower bound information becomes useless. This will make ALGORITHM 2 no different from a pure cutting plane approach. The indefinite QP is a type of very difficult problem because it was demonstrated that the indefinite QP is NP-hard even with only one negative eigenvalue, [17]. However,

according to the computational results obtained, we observed that they overall are quite encouraging. Most of the problems are solved within the first three iterations and the lower bound information obtained from (9) actually takes effects.

7 Further Research

For ALGORITHM 2 presented in this paper, some additional research directions are suggested. In calculating a tight lower bound, other approaches exist. For example, LINGO can generate a very tight lower bound (often it is the global optimum) even though its best objective value always remains far from the lower bound for a very long period. The Reformulation Linearization Technique (RLT) in [20, 21] is also a promising method in the sense that RLT can generate feasible points as well as lower bounds for BLP problems. Moreover, it has been proved that its lower bounds are at least as good as those generated by [1, 2]. However, the main drawback is that the amount of work required to construct the necessary matrix increases very rapidly as the number of variables and constraints grow due to the combinatorial number of cross products that must be considered. Therefore, RLT is not particularly attractive for our BLP problem which contains so many lower and upper bounds, although it would be interesting to test its relative merits for different problem sizes. A worthwhile direction would be to search for a tighter lower bound, than the one proposed herein, that is relatively inexpensive to compute.

The BLP problem in DELTA is very special; i.e., it only includes $x_i y_i$, which makes the matrix C in (3) simply possess diagonal entries with $+1$ and -1. However, as shown in [2], it is also possible to handle the case where the diagonal entries are arbitrary real numbers, thus creating weighted utility objective functions.

Suppose $b \in R^n$ and let $B = diag(b)$. Then the convex envelope of B is:

$$
\begin{aligned}
&Vex_{\Omega_i}[b_i x_i y_i] = \max\{b_i \varphi_i^1(x_i y_i), b_i \varphi_i^2(x_i y_i)\}, \\
&\text{where} \\
&\varphi_i^1(x_i y_i) = \begin{cases} m_i x_i + l_i y_i - l_i m_i & \text{if} \quad b_i > 0 \\ M_i x_i + l_i y_i - l_i M_i & \text{if} \quad b_i \le 0 \end{cases}, \\
&\varphi_i^2(x_i y_i) = \begin{cases} M_i x_i + L_i y_i - L_i M_i & \text{if} \quad b_i > 0 \\ m_i x_i + L_i y_i - L_i m_i & \text{if} \quad b_i \le 0 \end{cases}
\end{aligned} \tag{10}
$$

Moreover, the convex envelope of $x_i y_i$ can also be extended to $x_i y_j$ where $i \neq j$, and thereby underestimate arbitrary bilinear objective functions.

References

[1] Al-Khayyal, F.A. and Falk J.E. "Jointly Constrained Biconvex Programming", Mathematics of Operations Research, 8:273-286, 1983.

[2] Al-Khayyal, F.A. "Jointly Constrained Bilinear Programs and Related Problems: An Overview", Computers & Mathematics with Application, Vol.19, No.11:53-62, 1990.

[3] Balas, E. "Intersection Cuts - a New Type of Cutting Planes for Integer Programming", Operations Research 19:19-39, 1971.

[4] Danielson, M. *Computational Decision Analysis*, Doctoral Thesis, Department of Computer and Systems Sciences, Stockholm University and Royal Institute of Technology, 1997.

[5] Danielson, M. "Generalized Evaluation in Decision Analysis", in press, to appear in European Journal of Operational Research, 2004.

[6] Danielson, M. and Ekenberg, L. "A Framework for Analysing Decisions under Risk", European Journal of Operational Research, Vol.104(3):474-484, 1998.

[7] Danielson, M. and Ekenberg, L. "Multi-criteria Evaluation of Decision Trees", Proceedings of the 5th International Conference of the Decision Sciences Institute, 1999.

[8] Danielson, M. Ekenberg, L. Johansson, J. and Larsson, A. "The *Decide*IT Decision Tool", Proceedings of ISIPTA-2003, 2003.

[9] Ding, X.S. Danielson, M. and Ekenberg, L. "Non-linear Programming Solvers for Decision Analysis Support Systems", Proceedings of International Conference on Operations Research (OR 2003), pp.475-482, Springer-Verlag, 2004.

[10] Ding, X.S. Ekenberg, L. and Danielson, M. "A Fast Bilinear Optimization Algorithm", submitted, 2003.

[11] Ekenberg, L. Boman, M. and Linneroth-Bayer, J. "General Risk Constraints", Journal of Risk Research, 4(1):31-47, 2001.

[12] Ekenberg, L. Brouwers, L. Danielson, M. Hansson, K. Johansson, J. Riabacke, A. and Vári A. "Simulation and Analysis of Three Flood Management Strategies", IIASA Interim Report, IR-03-003, Laxenburg, Austria, 2003.

[13] Gallo, G. and Ülkücü, A. "Bilinear Programming: an Exact Algorithm", Mathematical Programming 12:173-194, 1977.

[14] Gill, P.E. Murray, W. and Saunders, M.A. "User's Guide for SQOPT 5.3: A Fortran Package for Large-scale Linear and Quadratic Programming", Report NA 97-4, Department of Mathematics, University of California, San Diego, USA, 1997.

[15] Horst, R. and Pardalos, P.M. *Handbook of Global Optimization*, Kluwer, Dordrecht, 1995.

[16] Konno, H. "Bilinear Programming", Parts I and II, Technical Report No. 71-9 and 71-10, Operations Research House, Stanford University, USA, 1971.

[17] Pardalos, P.M. and Vavasis, S.A. "Quadratic Programming with One Negative Eigenvalue Is NP-hard", Journal of Global Optimization, 1:15-23, 1991.

[18] Pardalos, P.M. and Romeijn, H.E. *Handbook of Global Optimization Volume 2*, Kluwer Academic Publishers, the Netherlands, 2002.

[19] Ritter, K. "A Method for Solving Maximum-Problems with a Nonconcave Quadratic Objective Function", Z. Wahrscheinlichkeitstheorie verw. Geb. 4:340-351, 1966.

[20] Sherali, H.D. and Adams, W.P. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, Kluwer Academic Publishers, Dordrecht/Boston/London, 1999.

[21] Sherali, H.D. and Alameddine, A. "A New Reformulation Linearization Algorithm for Bilinear Programming Problems", Journal of Global Optimization, Vol.2:379-410, 1992.

[22] Sherali, H.D. and Shetty, C.M. "A Finitely Convergent Algorithm for Bilinear Programming Problems Using Polar Cuts and Disjunctive Face Cuts", Mathematical Programming, Vol.19:14-31, 1980.

[23] Shetty, C.M. and Sherali, H.D. "Rectilinear Distance Location-Allocation Problem: A Simplex Based Algorithm", Proceedings of the International Symposium on Extremal Methods and Systems Analyses, Springer-Verlag, Vol.174:442-464, 1980.

[24] http://www.tomlab.biz/

[25] Tuy, H. "Concave Programming under Linear Constraints", Dokl. Akad. Naul SSR 159:32-35; English translation in Soviet Math. Dokl. 5:1437-1440, 1964.

[26] Vaish, H. and Shetty, C.M. "A Cutting Plane Algorithm for the Bilinear Programming Problem", Naval Reaearch Logistics Quarterly 24:83-94, 1977.

Topology Optimization of Navier–Stokes Equations

Anton Evgrafov

Chalmers University of Technology, Göteborg, SE-412 80, Sweden

We consider the problem of optimal design of flow domains for Navier–Stokes flows in order to minimize a given performance functional. We attack the problem using topology optimization techniques, or control in coefficients, which are widely known in structural optimization of solid structures for their flexibility, generality, and yet ease of use and integration with existing FEM software. Topology optimization rapidly finds its way into other areas of optimal design, yet until recently it has not been applied to problems in fluid mechanics. The success of topology optimization methods for the minimal drag design of domains for Stokes fluids has lead to attempts to use the same optimization model for designing domains for incompressible Navier–Stokes flows. We show that the optimal control problem obtained as a result of such a straightforward generalization is ill-posed, at least if attacked by the direct method of calculus of variations. We illustrate the two key difficulties with simple numerical examples and propose changes in the optimization model that allow us to overcome these difficulties. Namely, to deal with impenetrable inner walls that may appear in the flow domain we slightly relax the incompressibility constraint as typically done in penalty methods for solving the incompressible Navier–Stokes equations. In addition, to prevent discontinuous changes in the flow due to very small impenetrable parts of the domain that may disappear, we consider so-called filtered designs, that has become a "classic" tool in the topology optimization toolbox. Technically, however, our use of filters differs significantly from their use in the structural optimization problems in solid mechanics, owing to the very unlike design parametrizations in the two models. We rigorously establish the well-posedness of the proposed model and then discuss related computational issues.

I. Introduction

The optimal control of fluid flows has long been receiving considerable attention by engineers and mathematicians, owing to its importance in many applications involving fluid related technology.[1,2] According to a well-established classification in structural optimization (see Ref. 3, page 1), the absolute majority of works dealing with optimal design of flow domains fall into the category of shape optimization. (See the bibliographical notes [2] in Ref. 3 for classic references in shape optimization.) In the framework of *shape optimization*, the optimization problem formulation can be stated as follows: choose a flow domain out of some family so as to maximize an associated performance

functional. The family of domains considered may be as rich as that of all open subsets of a given set satisfying some regularity criterion,[4] or as poor as the ones obtained from a given domain by locally perturbing some part of the boundary in a Lipschitz manner.[5,6,7] Unfortunately, it is typically only the problems in the latter group that can be attacked numerically. On the other hand, *topology optimization* (or, control in coefficients) techniques are known for their flexibility in describing the domains of arbitrary complexity (e.g., the number of connected components need not to be bounded), and at the same time require relatively moderate efforts from the computational part. In particular, one may completely avoid remeshing the domain as the optimization algorithm advances, which eases the integration with existing FEM codes, and simplifies and speeds up sensitivity analysis.

While the field of topology optimization is very well established for optimal design of solids and structures, surprisingly little work has been done for optimal design of fluid domains. Borrvall and Petersson[8] considered the optimal design of flow domains for minimizing the total power of the incompressible Stokes flows, using inhomogeneous porous materials with a spatially varying Darcy permeability tensor, under a constraint on the total volume of fluid in the control region. Later, this approach has been generalized to include both limiting cases of the porous materials, i.e., pure solid and pure flow regions have been allowed to appear in the design domain as a result of the optimization procedure.[9] (We also cite the work of Klarbring et al.,[10] which however studies the problem of optimal design of flow networks, where design and state variables reside in finite-dimensional spaces; in some sense this is an analogue of truss design problems if one can carry over the terminology and ideas from the area of optimal design of structures and solids.)

Unfortuntely, applications of the Stokes flows are rather limited, while the Navier–Stokes equations are now regarded as the universal basis of fluid mechanics.[11] Therefore, it has been suggested that the optimization model proposed in Ref. 8 (with straightforward modifications), in particular the same design parametrization should be used for the topology optimization of the incompressible Navier–Stokes equations.[12] Essentially, in this model we control the Brinkman-type equations including the nonlinear convection term[13] (which will be referred to as nonlinear Brinkman equations in the sequel) by varying a coefficient before the zeroth order velocity term. Setting the control coefficient to zero is supposed to recover the Navier–Stokes equations; at the same time, infinite values of the coefficient are supposed to model the impenetrable inner walls in the domain. In Section III we illustrate the difficulties inherent in this approach, namely that the design-to-flow mapping is not closed, leading to ill-posed control problems.

It turns out that if we employ the idea of *filter*[14,15] (which has become quite a standard technique in topology optimization, see Refs. 16, 17 for the rigorous mathematical treatment) *in addition* to relaxing the incompressibility constraint (which is unique to the topology optimization of fluids) we can establish the continuity of the resulting design-to-flow mapping, and therefore the existence of optimal designs for a great variety of design functionals; this is discussed in Section IV. Not going into details yet, we comment that our use of filters significantly differs from the traditional one in the topology optimization. Namely, not only do we use filters to forbid small features from appearing in our designs and thus to transform weak(-er) design convergence into a strong(-er) one (cf. Proposition 5), but also to verify certain growth conditions near impenetrable walls [see inequality (4)], which later guarantees the embedding of certain weighted Sobolev spaces into classic ones and finally allows us to prove the continuity of design-to-flow mappings in Section V. The existence of optimal designs, formally established in Section VI,

is an easy corollary of the continuity of the design-to-flow mappings.

Some computational techniques are introduced in Section VII. Namely, we discuss a standard topic of approximating the topology optimization problems with so-called sizing optimization problems (also known as "ε-perturbation"), which in our case reduces to approximation of the impenetrable walls with materials of very low permeability, and then we touch upon techniques aimed at reducing the amount of porous material in the optimal design. We conclude the paper by discussing possible extensions of the presented results, open questions, and further research topics in Section VIII.

II. Prerequisites

A. Notation

Let Ω be a connected bounded domain of $\mathbb{R}^d$, $d \in \{\,2, 3\,\}$ with a Lipschitz continuous boundary Γ. In this domain we would like to control the nonlinear Brinkman equations[13] with the prescribed flow velocities $\boldsymbol{g}$ on the boundary, and forces $\boldsymbol{f}$ acting in the domain by adjusting the inverse permeability α of the medium occupying Ω, which depends on the control function ρ:

$$\begin{cases} \left.\begin{aligned} -\nu\boldsymbol{\Delta}\boldsymbol{u} + \boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u} + \alpha(\rho)\boldsymbol{u} + \boldsymbol{\nabla}p = \boldsymbol{f}, \\ \operatorname{div}\boldsymbol{u} = 0 \end{aligned}\right\}, & \text{in } \Omega, \\ \boldsymbol{u} = \boldsymbol{g}, & \text{on } \Gamma. \end{cases} \tag{1}$$

In the system (1), $\boldsymbol{u}$ is the flow velocity, p is the pressure, and ν is the kinematic viscosity. Of course, owing to the incompressibility of $\boldsymbol{u}$, the function $\boldsymbol{g}$ must satisfy the compatibility condition

$$\int_\Gamma \boldsymbol{g}\cdot\boldsymbol{n} = 0, \tag{2}$$

where $\boldsymbol{n}$ denotes the outward unit normal. If $\alpha(\rho(\boldsymbol{x})) = +\infty$ for some $\boldsymbol{x} \in \Omega$, we simply require $\boldsymbol{u}(\boldsymbol{x}) = 0$ in the first equation of the system (1).

Our control set $\mathcal{H}$ is defined as follows:

$$\mathcal{H} = \{\,\rho \in L^\infty(\Omega) \mid 0 \le \rho \le 1, \text{a.e. in } \Omega, \int_\Omega \rho \le \gamma|\Omega|\,\},$$

where $0 < \gamma < 1$ is the maximal volume fraction that can be occupied by the fluid. Every element $\rho \in \mathcal{H}$ describes the scaled Darcy permeability tensor of the medium at a given point $\boldsymbol{x} \in \Omega$ in the following (informal) way: $\rho(\boldsymbol{x}) = 0$ corresponds to zero permeability at $\boldsymbol{x}$ (i.e., solid, which does not permit any flow at a given point), while $\rho(\boldsymbol{x}) = 1$ corresponds to infinite permeability (i.e., 100% flow region; no structural material is present). Formally, we relate the permeability α^{-1} to ρ using a convex, decreasing, and nonnegative function[8,9] $\alpha : [0, 1] \to \mathbb{R}_+ \cup \{+\infty\}$, defined as

$$\alpha(\rho) = \rho^{-1} - 1.$$

In the rest of the paper we will use the symbol χ_A for $A \subset \Omega$ to denote the characteristic function of A: $\chi_A(\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in A$; $\chi_A(\boldsymbol{x}) = 0$ otherwise.

B. Variational formulation

To state the problem in a more analytically suitable way and to incorporate the special case $\alpha = +\infty$ into the first equation of the system (1), we introduce a weak formulation

of the equations. Let us consider the sets of admissible flow velocities:

$$\mathcal{U} = \{\, \boldsymbol{v} \in H^1(\Omega) \mid \boldsymbol{v} = \boldsymbol{g} \ \text{ on } \Gamma \,\},$$
$$\mathcal{U}_{\text{div}} = \{\, \boldsymbol{v} \in \mathcal{U} \mid \operatorname{div} \boldsymbol{v} = 0, \text{weakly in } \Omega \,\}.$$

Let $\mathcal{J}^{\mathcal{S}} : \mathcal{U} \to \mathbb{R}$ denote the potential power of the viscous flow:

$$\mathcal{J}^{\mathcal{S}}(\boldsymbol{u}) = \frac{\nu}{2} \int_\Omega \boldsymbol{\nabla u} \cdot \boldsymbol{\nabla u} - \int_\Omega \boldsymbol{f} \cdot \boldsymbol{u}.$$

Let us further define the additional power dissipation $\mathcal{J}^{\mathcal{D}} : \mathcal{H} \times \mathcal{U} \to \mathbb{R} \cup \{+\infty\}$, due to the presence of the porous medium (we use the standard convention $0 \cdot +\infty = 0$):

$$\mathcal{J}^{\mathcal{D}}(\rho, \boldsymbol{u}) = \frac{1}{2} \int_\Omega \alpha(\rho) \boldsymbol{u} \cdot \boldsymbol{u}.$$

Finally, let $\mathcal{J}(\rho, \boldsymbol{u}) = \mathcal{J}^{\mathcal{S}}(\boldsymbol{u}) + \mathcal{J}^{\mathcal{D}}(\rho, \boldsymbol{u})$ denote the total power of the Brinkman flow. Then, the requirement "$\alpha(\rho) = +\infty \implies \boldsymbol{u} = 0$" is automatically satisfied if $\mathcal{J}^{\mathcal{D}}(\rho, \boldsymbol{u}) < +\infty$.

We will use epi-convergence of optimization problems as a main theoretical tool in the subsequent analysis, thus it is natural to study the following variational formulation[9] for Darcy-Stokes flows [i.e., obtained by neglecting the convection term $\boldsymbol{u} \cdot \boldsymbol{\nabla u}$ in the system (1)]: for $\boldsymbol{f} \in L^2(\Omega)$, compatible $\boldsymbol{g} \in H^{1/2}(\Gamma)$, and $\rho \in \mathcal{H}$, find $\boldsymbol{u} \in \mathcal{U}_{\text{div}}$ such that

$$\boldsymbol{u} \in \operatorname*{argmin}_{\boldsymbol{v} \in \mathcal{U}_{\text{div}}} \mathcal{J}(\rho, \boldsymbol{v}).$$

Naturally, taking convection into account, this can be generalized to the following fixed point-type formulation of the system (1): for $\boldsymbol{f} \in L^2(\Omega)$, compatible $\boldsymbol{g} \in H^{1/2}(\Gamma)$, and $\rho \in \mathcal{H}$ find $\boldsymbol{u} \in \mathcal{U}_{\text{div}}$ such that

$$\boldsymbol{u} \in \operatorname*{argmin}_{\boldsymbol{v} \in \mathcal{U}_{\text{div}}} \left\{ \mathcal{J}(\rho, \boldsymbol{v}) + \int_\Omega (\boldsymbol{u} \cdot \boldsymbol{\nabla u}) \cdot \boldsymbol{v} \right\}. \tag{3}$$

III. Problems with the existing approach

When we allow impenetrable walls to appear and to disappear in the design domain, we create two particular types of difficulties, each related to a corresponding change in topology (see Subsections A and B). We note that in the "sizing" case, which can be modeled by introducing an additional design constraint $\rho \geq \varepsilon$, a.e. in Ω (for some small $\varepsilon > 0$) these difficulties do not appear. (In fact, it is an easy exercise to verify that under such circumstances the design-to-flow mapping is closed w.r.t. strong convergence of designs, e.g., in $L^1(\Omega)$, and $H^1(\Omega)$-weak convergence of flows.) Such a distinct behavior of the sizing and topology optimization problems may indicate that the former is not a useful approximation of the latter in this case.

A. Disappearing walls

For the sake of simplicity, in this subsection we assume that the objective functional in our control problem (which is not formally stated yet) is the power $\mathcal{J}$ of the incompressible Navier–Stokes flow. This functional is interesting from at least two points of view. Firstly, in many cases the resulting control problem is equivalent to the minimization of the drag force or pressure drop, which is very important in engineering applications.[8] Secondly, while it is intuitively clear that impenetrable inner walls of vanishing thickness change

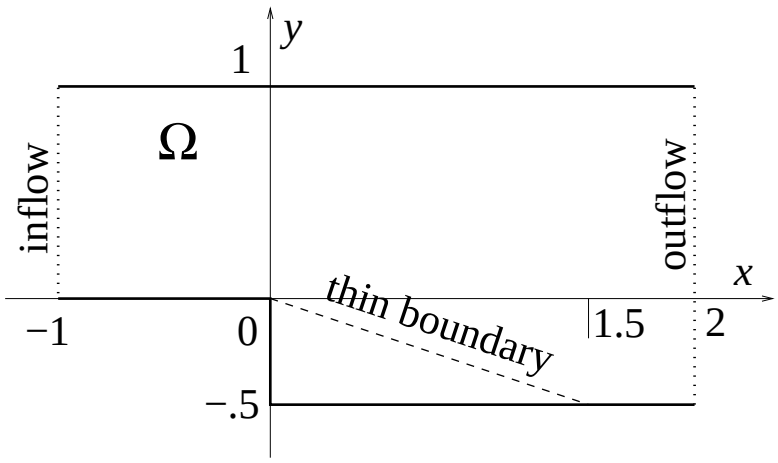


Figure 1. Flow domain for the backstep flow.

the flow in a discontinuous way, for the Stokes flows the total potential power is lower semi-continuous w.r.t. such changes, which allows us to apply the Weierstrass theorem and ensure the existence of optimal designs (see Theorem 3.3 in Ref. 9). In this subsection we consider two examples illustrating the discontinuity of the flow as well as non-lower semicontinuity of the power functional in the case of the incompressible Navier–Stokes equations; this means that the corresponding control problem of minimizing the potential power is ill-posed, at least from the point of view of the direct method of calculus of variations.

Example 1 (Infinitely thin wall). We consider a variant of the backstep flow with $\nu = 1.0 \cdot 10^{-3}$ (which corresponds to the Reynolds number $\mathrm{Re} = 1000$), as shown in Fig. 1. We specify $\boldsymbol{u}$ on the inflow boundary to be $(0.25 - (y - 0.5)^2, 0.0)^t$, on the outflow boundary we require $u_y = 0$ as well as $p = 0$; on the rest of the boundary the no-slip condition $\boldsymbol{u} = \boldsymbol{0}$ is assumed. We consider a sequence of the domains containing a thin but impenetrable wall of vanishing thickness (as shown in Fig. 1 by dashed line). The limiting domain is the usual backstep shown with the solid line. Direct numerical computation in Femlab (see Fig. 2 showing the flows) shows that for the domains with thin wall we have $\mathcal{J} \approx 0.8018$, while for the limiting domain $\mathcal{J} \approx 0.8263$. This demonstrates the non-lower semicontinuity of the total power functional in the case of incompressible Navier–Stokes equations.

We note that while the "jump" of the power functional may seem negligible in this example, other examples may be constructed where this jump is much bigger.

It may be argued that in the example above the thin wall may be substituted by the complete filling of the resulting isolated subdomain with impenetrable material, and the following example is more peculiar and demonstrates that we can control the behavior of the Navier–Stokes flow with an infinitesimal amount of material. It is interesting to note that the example is based on the construction of Allaire,[13] which in some sense is "opposite" to our design parametrization. Namely, we try to control the Navier–Stokes equations by adjusting the coefficients in the nonlinear Brinkman equations, while the sequence of perforated domains considered in Example 2 has been used to obtain the nonlinear Brinkman equations starting from the Navier–Stokes equations in a periodically perforated domain as a result of the homogenization process.

Example 2 (Perforated domains with tiny holes). We assume that the boundary Γ is *smooth* and impenetrable (i.e., the homogeneous boundary conditions $\boldsymbol{g} = \boldsymbol{0}$ hold),

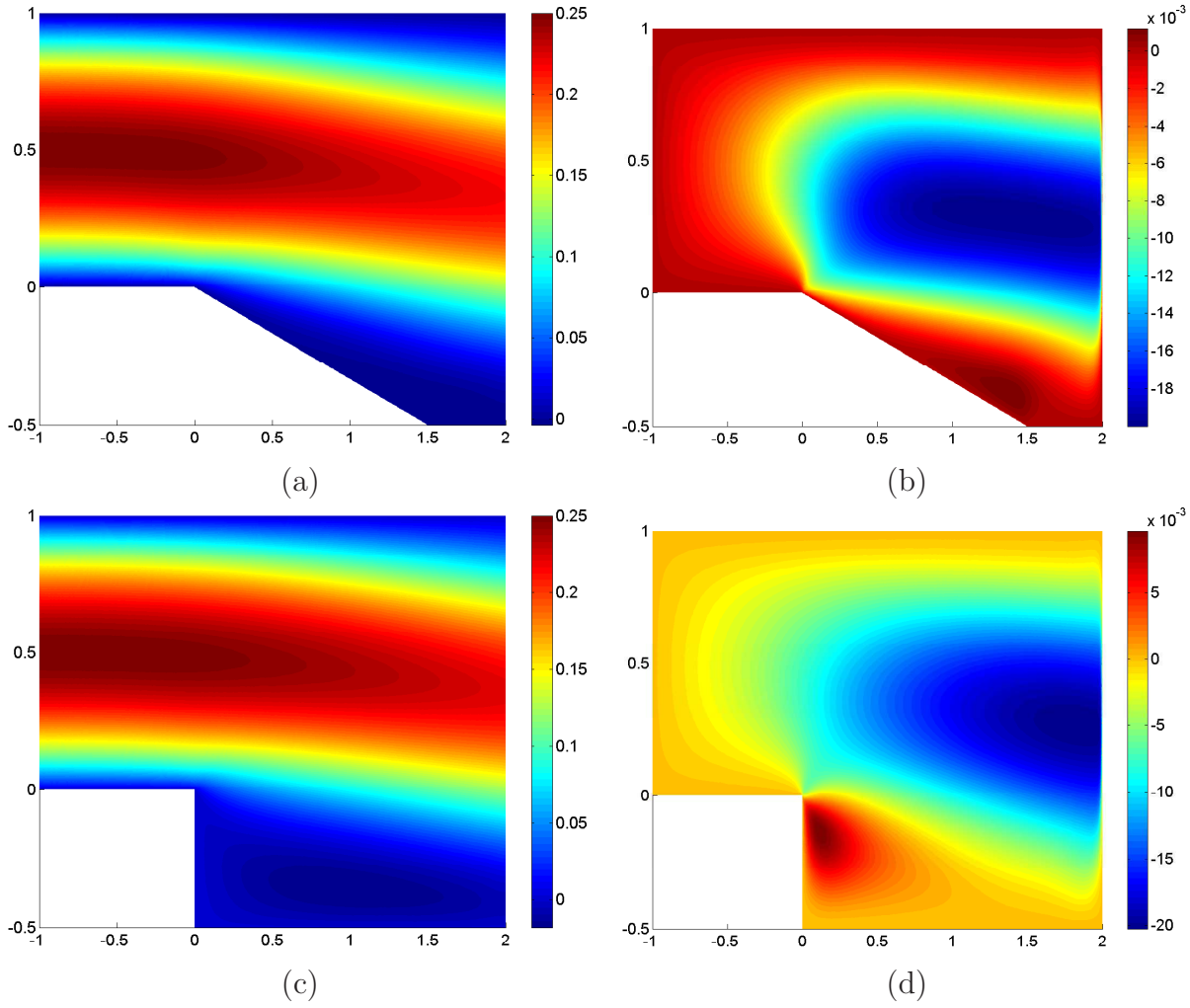


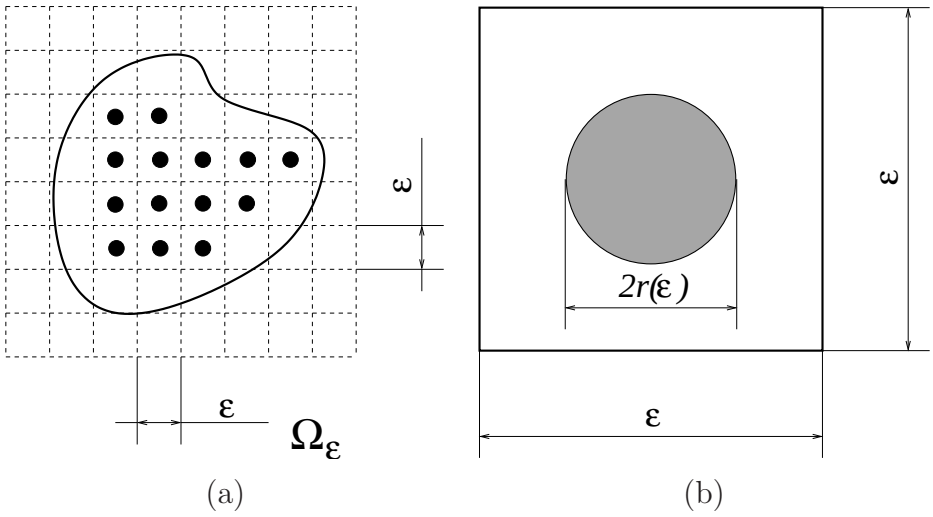
Figure 2. Backstep flow: Example 1. (a), (b): x- and y-components, respectively, of the flow velocity when the impenetrable wall has arbitrary but positive thickness (only the part of the domain with nontrivial flow is shown); (c), (d): x- and y-components, respectively, of the flow velocity as the impenetrable wall disappears. *Note the different color scales.*

and that the viscosity ν is large enough relatively to the force $\boldsymbol{f}$ to guarantee the existence of a unique solution to the Navier–Stokes system in Ω. Let Ω^ε denote a perforated domain, obtained from Ω by taking out spheres of radius $r_d(\varepsilon)$ with centers $\varepsilon\mathbb{Z}^d$, where $\lim_{\varepsilon\to+0} r_d(\varepsilon)/\varepsilon = 0$; see Fig. 3. Let $(\widetilde{\boldsymbol{u}}^\varepsilon, \widetilde{p}^\varepsilon)$ denote a solution to the Navier–Stokes problem inside Ω^ε with homogeneous boundary conditions $\widetilde{\boldsymbol{u}}^\varepsilon = \boldsymbol{0}$ on $\partial\Omega^\varepsilon$. We extend $\widetilde{\boldsymbol{u}}^\varepsilon$ onto the whole Ω by setting it to zero inside each sphere; we further denote by $\boldsymbol{u}^\varepsilon$ this extended solution. For every small $\varepsilon > 0$ it holds that $\boldsymbol{u}^\varepsilon$ solves the problem (3) for $\rho^\varepsilon = \chi_{\Omega^\varepsilon}$. Allaire[13] has shown that depending on the limit $C = \lim_{\varepsilon\to+0} r_d(\varepsilon)/\varepsilon^3$ for $d = 3$, or $C = \lim_{\varepsilon\to+0} -\varepsilon^2 \log(r_d(\varepsilon))$ for $d = 2$, there are three limiting cases:

$C = 0$: $\{\boldsymbol{u}^\varepsilon\}$ converges strongly in $H^1(\Omega)$ towards the solution to the Navier–Stokes problem in the unperforated domain Ω, i.e., the solution to the problem (3) corresponding to $\rho = 1$ (see Theorem 3.4.4 in Ref. 18);

$C = +\infty$: $\{\boldsymbol{u}^\varepsilon\}$ converges towards 0 strongly in $H^1(\Omega)$ (in fact, there is more information about $\{\boldsymbol{u}^\varepsilon\}$ available, see Theorem 3.4.4 in Ref. 18);

$0 < C < +\infty$: $\{\boldsymbol{u}^\varepsilon\}$ converges weakly in $H^1(\Omega)$ towards the solution to the nonlinear Brinkman problem in the unperforated domain Ω, i.e., the solution of the problem (3) corresponding to $\rho = \sigma$, for a computable constant



(a) (b)

Figure 3. The perforated domain (a) **and a periodic cell** (b)**.**

$\sigma(d, \nu, C) > 0$ (see Main Theorem in Ref. 13).

We note that in all three cases the sequence of designs $\{\rho^\varepsilon\}$ strongly converges to zero in $L^1(\Omega)$, while only in the case $C = 0$ the corresponding sequence of flows converges to the "correct" flow. As for the other two cases, we can either completely stop ($C = +\infty$) or just slow ($0 < C < +\infty$) the flow using only infinitesimal amounts of structural material (recall that $r_d(\varepsilon)/\varepsilon \to +0$). Moreover, the sequence of perimeters of ρ^ε converges to zero, and therefore the perimeter constraint cannot enforce the convergence of flows in this case (contrary to the situation in linear elasticity, see p. 31 in Ref. 3). In the same spirit, the regularized intermediate density control method considered by Borrvall and Petersson[19] classifies the designs ρ^ε as regular for all enough small $\varepsilon > 0$ (since they are indeed close to a regular design $\rho \equiv 0$ in the strong topology of $L^p(\Omega)$, $1 \le p < \infty$); thus the latter method also fails to recognize the pathological cases illustrated in the present example.

B. Appearing walls

Walls that appear in the domain as a result of the optimization process may break the connectivity of the flow domain (or some parts of it), so that the incompressible Navier–Stokes system may not admit any solutions in the limiting domain (resp., some parts of it). While obtaining such results may seem to be a failure of the optimization procedure, completely stopping the flow might be interesting (or even optimal) with respect to some engineering design functionals.

The following example is purely artificial and its only purpose is to demonstrate the possible non-closedness of the design-to-flow mapping when new walls appear in the domain. It essentially repeats Example 2.1 in Ref. 9, but we include it here for convenience of the reader.

Example 3 (Domain with diminishing permeability). Let $\Omega = (0, 1)^2$, $\boldsymbol{g} \equiv (1, 0)^t$, and $\boldsymbol{f} \equiv \boldsymbol{0}$. Let further $\rho_k \equiv 1/k$ in Ω, $k = 1, 2, \ldots$, $\rho \equiv 0$ in Ω, so that $\rho_k \to \rho$, strongly in $L^\infty(\Omega)$ as $k \to \infty$. Then, $\boldsymbol{u} \equiv (1, 0)^t$ is a solution of the problem (3) for all $k = 1, 2, \ldots$; clearly, $(\rho_k, \boldsymbol{u}) \to (\rho, \boldsymbol{u})$, strongly in $L^\infty(\Omega) \times H^1(\Omega)$. At the same time, it is not difficult to verify that the problem (3) has no solutions for the limiting design ρ, which means that the design-to-flow mapping is not closed even in the strong topology of $L^\infty(\Omega) \times H^1(\Omega)$!

The problem related to the appearence of walls completely stopping the flow in some

domains has been solved for Stokes flows by (implicitly) introducing an additional constraint $\mathcal{J}(\rho, \boldsymbol{u}) \leq C$, for a suitable constant C. Owing to the coercivity of $\mathcal{J}$ on $H_0^1(\Omega)$, this keeps the flows in some bounded set. However, in view of the non-lower semicontinuity of the power functional for the Navier–Stokes flows (see Example 1), this set is not necessarily closed, making the problems with appearing walls much more severe in the present case.

We consider the next example in some detail, even though it is quite similar to the previous one, because we will return to it later in Subsection B of Section IV.

Example 4 (Channel with a porous wall). We consider a channel flow at Reynolds number $\mathrm{Re} = 1000$ ($\nu = 1.0 \cdot 10^{-3}$) through a wall made of porous material with vanishing permeability appearing in the middle of the channel (see Fig. 4). We specify $\boldsymbol{u}$ on the

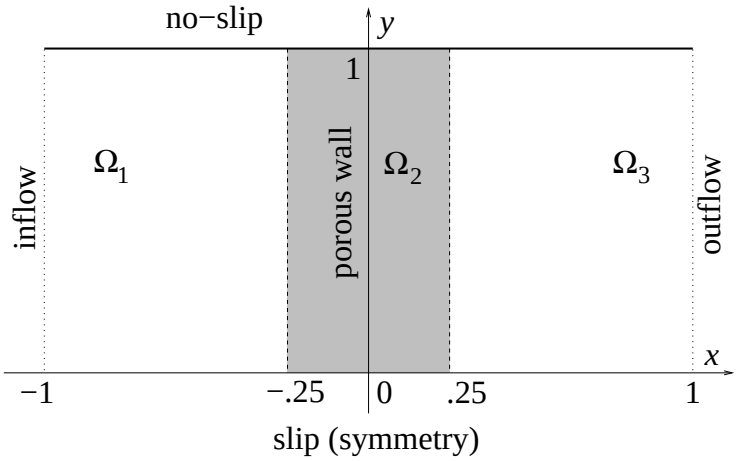


Figure 4. Flow domain of Example 4.

inflow boundary to be $(1 - y^2, 0.0)^t$, on the outflow boundary we require $u_y = 0$ as well as $p = 0$; on the rest of the boundary the no-slip condition $\boldsymbol{u} = \boldsymbol{0}$ is assumed except that on the "lower" edge we have slip (i.e., only $u_y = 0$) due to the symmetry.

We choose ρ so that $\alpha(\rho) = 0$ on $\Omega_1 \cup \Omega_3$ and $\alpha(\rho) = \alpha$ on Ω_2, where α assumes values 1.0, $1.0 \cdot 10^2$, $1.0 \cdot 10^4$, $+\infty$. The corresponding flows (calculated in Femlab) are shown in Fig. 5; the incompressible Navier–Stokes problem in the last (limiting as $\alpha \to +\infty$) domain admits no solutions.

To summarize, even though the sequence of designs $\rho_\alpha \to \chi_{\Omega_1 \cup \Omega_3}$, strongly in $L^\infty(\Omega)$, the corresponding sequence of flows does not converge to the flow corresponding to the limiting design, simply because the latter does not exist.

IV. Proposed solutions to the difficulties outlined

Difficulties inherent in the straightforward generalization of the methodology proposed by Borrvall and Petersson[8] for Stokes flows to incompressible Navier–Stokes flows have been outlined in Section III. One possible solution, which allows us to avoid these difficulties, is simply to forbid topological changes and to perform sizing optimization, interpreting optimal designs as distributions of porous materials with spatially varying permeability.[13,18,20] As it has already been mentioned the resulting designs may or may not accurately describe the domains obtained by substituting the materials with high permeability by void, and those with low permeability by impenetrable walls. Furthermore, if we decide to keep the porous material, it is questionable whether such designs

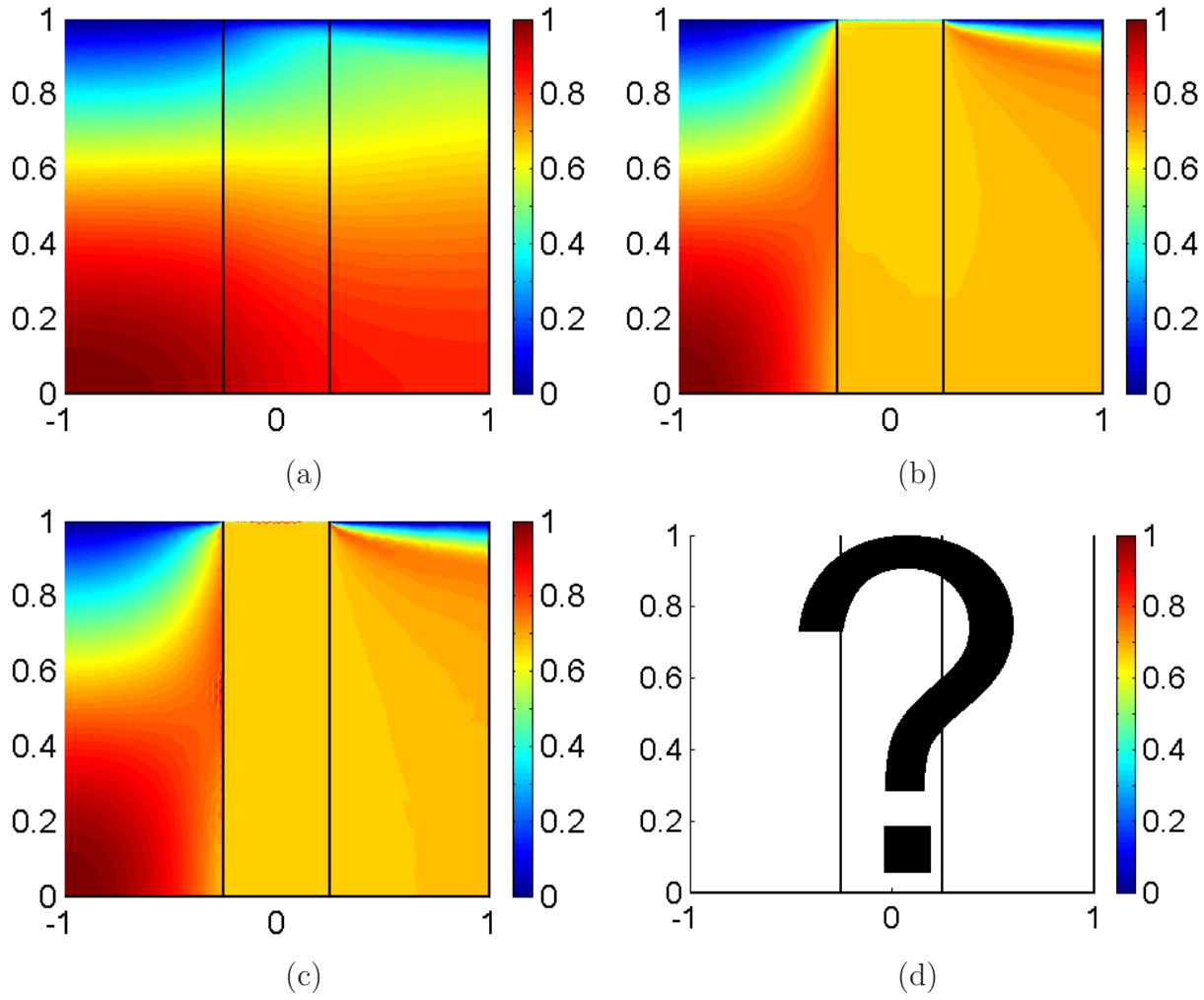


Figure 5. Incompressible flow through the porous wall: (a) $\alpha = 1.0$, (b) $\alpha = 1.0 \cdot 10^2$, (c) $\alpha = 1.0 \cdot 10^4$, (d) $\alpha = +\infty$.

can be easily manufactured and thus it is unclear whether they are "better" from the engineering point of view. Thus we do not employ this approach but instead try to slightly modify the design parametrization as well as the underlying state equations with the ultimate goal to rigorously obtain a closed design-to-flow mapping while maintaining a clear engineering/physical meaning of our optimization model.

A. Filters in the topology optimization

In both examples in Subsection A of Section III we constructed the sequences of designs having very small details, which disappear in the limit. Using the notion of a filter[14,15] we can control the minimal scale of our designs; we will employ this technique, which has become quite standard in topology optimization of linearly elastic materials.[3]

Following Bourdin,[16] and Bruns and Tortorelli,[17] we define a *filter* $F : \mathbb{R}^d \to \mathbb{R}$ *of characteristic radius* $R > 0$ to be a function verifying the following properties:

$$F \in C^{0,1}(\mathbb{R}^d), \qquad \operatorname{supp} F \Subset B_R, \qquad \operatorname{supp} F \text{ is convex},$$
$$F \geq 0 \text{ in } B_R, \qquad \int_{B_R} F = 1,$$

where B_R denotes the open ball of radius R centered in origo. We denote the convolution

product by a $*$ sign, i.e.

$$(F * \rho)(\boldsymbol{x}) = \int_{\mathbb{R}^d} F(\boldsymbol{x} - \boldsymbol{y})\rho(\boldsymbol{y})d\boldsymbol{y}.$$

Owing to the Lipschitz continuity of F, $F * \rho$ is a continuous function.

In order to compute the convolution between the filter and a given design ρ the latter must be defined not only on Ω, but also on the whole space $\mathbb{R}^d$. Therefore, in the sequel we consider the following redefined design domain:

$$\mathcal{H} = \{\, \rho \in L^\infty(\mathbb{R}^d) \cap L^1(\mathbb{R}^d) \mid 0 \le \rho \le 1, \text{a.e. in } \mathbb{R}^d, \int_{\mathbb{R}^d} \rho \le V \,\},$$

for a given $V > 0$.

One of the consequences of the fact that F is Lipschitz continuous in $\mathbb{R}^d$ and not just in B_R is that the following important *growth condition* is verified:

$$(F * \chi_{\mathbb{R}^d \setminus \operatorname{supp} F})(\boldsymbol{x}) \le C|\boldsymbol{x}|^2, \tag{4}$$

as $|\boldsymbol{x}| \to 0$, for some appropriate constant $C > 0$, which implies that $\alpha((F * \rho)(\cdot))$ grows at least as fast as $\operatorname{dist}^{-2}(\cdot, \{\, F * \rho = 0 \,\})$ arbitrarily near to impenetrable walls. It is this condition that is the key ingredient in the proof of closedness theorems.

For notational convenience we set $\mathcal{J}^F(\rho, \boldsymbol{u}) = \mathcal{J}(F * \rho, \boldsymbol{u})$. As a consequence of the introduction of the filter, we can demonstrate the following simple claim, which translated to normal language says that impenetrable walls cannot disappear in the limit. In the following Proposition, Lim sup is understood in the sense of Painlevé-Kuratowski.

Proposition 5. *Consider an arbitrary sequence of designs $\{\rho_k\} \subset \mathcal{H}$, such that $\rho_k \rightharpoonup \rho$, weakly in $L^1_{\mathrm{loc}}(\mathbb{R}^d)$, for some $\rho \in \mathcal{H}$. Define a sequence $\{\Omega_0^k\}$ of subsets of Ω as*

$$\Omega_0^k = \{\, \boldsymbol{x} \in \Omega \mid (F * \rho_k)(\boldsymbol{x}) = 0 \,\},$$
$$\Omega_0^\infty = \{\, \boldsymbol{x} \in \Omega \mid (F * \rho)(\boldsymbol{x}) = 0 \,\}.$$

Then, $\operatorname{Lim\,sup}_{k\to\infty} \Omega_0^k \subset \Omega_0^\infty \cup \Gamma$.

Remark 6. The convergence of flow domains $\Omega \setminus \Omega_0^k$ induced by the weak convergence of designs (which implies strong convergence of filtered designs) can be compared to the convergence of domains in some topology defined for set convergence, e.g., the complementary Hausdorff topology. It is known, in general, that the latter topology is weaker (see, e.g., [21, Section 2.6.2]). However, such a comparison is not quite fair in the present situation, where the domains we deal with can be rather irregular (e.g., lie on two sides of their boundaries), and, more importantly, the domains in the sequence may have different connectivity compared to the "limiting" domain.

Later we will see that we need even stronger convergence of $\Omega_0^k \to \Omega_0^\infty$ to obtain closedness of the design-to-flow mappings.

The use of filtered designs $F * \rho$ in place of ρ in the problem (3) allows us to overcome the difficulties caused by disappearing walls. While we delay the formal statement of this fact until Section V, at this point we can consider an example that illustrates the effect of using filters.

Example 7 (Example 2 revisited). Consider an arbitrary filter F and a sequence of designs $\{\rho_\varepsilon\}$ defined in Example 2. Let for every $\varepsilon > 0$ extend the definition of ρ_ε (that has been defined only on Ω) by setting $\rho_\varepsilon(\boldsymbol{x}) = 1$ for all $\boldsymbol{x} \in (\Omega + \operatorname{supp} F) \setminus \Omega$, and $\rho_\varepsilon(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathbb{R}^d \setminus (\Omega + \operatorname{supp} F)$. Then, $F * \rho_\varepsilon \to 1$ as $\varepsilon \to +0$, uniformly in $\operatorname{cl}\Omega$, and the corresponding sequence of flows converges to a pure Navier–Stokes flow in the domain Ω (case $C = 0$ in Example 2).

B. Slightly compressible fluids

While it seems difficult to imagine a reasonable cure for Example 3, because the limiting flow must be zero on Ω with nonzero trace on Γ, we can at least try to get a closed design-to-flow mapping if impenetrable walls do not appear too close to the boundary with non-homogeneous Dirichlet conditions on velocity, as in Example 4. The difficulty in the latter example is that in our model the porous wall does not stop, or slow, the incompressible fluid while we use material with positive permeability. At the same time, the limiting domain does not permit any incompressible flow through it, because it is not connected.

We can solve this problem by relaxing the incompressibility requirement $\operatorname{div} \boldsymbol{u} = 0$ in the system (1) [of course, we do not need to require the compatibility condition (2) in this case]. For example, we may assume that the fluid is *slightly compressible*, i.e., choose a small $\delta > 0$ and let $\operatorname{div} \boldsymbol{u} + \delta p = 0$. In fact, it is known that for a fixed domain admitting an incompressible flow, the difference between the regular incompressible and slightly compressible flows is of order δ, i.e., we change model only slightly if δ is small enough. The slightly compressible Navier–Stokes equations are often used as approximations of incompressible ones in so-called *penalty algorithms* (see Chapter 5 in Ref. 22). On the other hand, with the gained maturity of mixed finite element methods, the incompressible system can be equally well solved to approximate the behavior of slightly compressible fluids.[23]

Whether one considers slightly compressible Navier–Stokes fluids to be the most suitable mathematical model of the underlying physical flow (see Remark 9) or just an accurate approximation of the incompressible Navier–Stokes equations, we make an assumption of slight compressibility because it allows us to achieve the ultimate goal of this paper: to obtain a closed design-to-flow mapping. Again, delaying the precise formulations until Section V, we revisit Example 4 to illustrate our point.

Example 8 (Example 4 revisited). We choose $\delta = 1.0 \cdot 10^{-3}$ and resolve the flow problem of Example 4 for $\alpha \in \{ 1.0, 1.0 \cdot 10^2, 1.0 \cdot 10^4, +\infty \}$. The corresponding flows (calculated in Femlab) are shown in Fig. 6; in contrast with the incompressible Navier–Stokes case we can see the convergence of flows as domains converge (i.e., as α increases) to a limiting flow, which exists in the compressible case. Note that for small values of α and δ the incompressible and the slightly compressible flows look similar.

Remark 9. It is known that the pseudo-constitutive relation $\operatorname{div} \boldsymbol{u} + \delta p = 0$ lacks an adequate physical interpretation for many important physical flows.[24] In particular, there is no physical pressure field compatible with the flow shown in Fig. 6 (d). On the other hand, the pseudo-constitutive relation resulting from the penalty method can still be used as a mathematical method of generating flows approximating those of incompressible viscous fluids. Moreover, the idea of relaxing the incompressibility contraint may also be useful for topology optimization in fluid *dynamics*, where the corresponding relation $\operatorname{div} \boldsymbol{u} + \delta dp/dt = 0$ is known to be physical.

V. Continuity of the design-to-flow mapping

A. Stokes flows

We start by showing the closedness of the design-to-flow mapping for slightly compressible Stokes flows with homogeneous boundary conditions, and then show the necessary modifications for the inhomogeneous boundary conditions. For the compressible Stokes

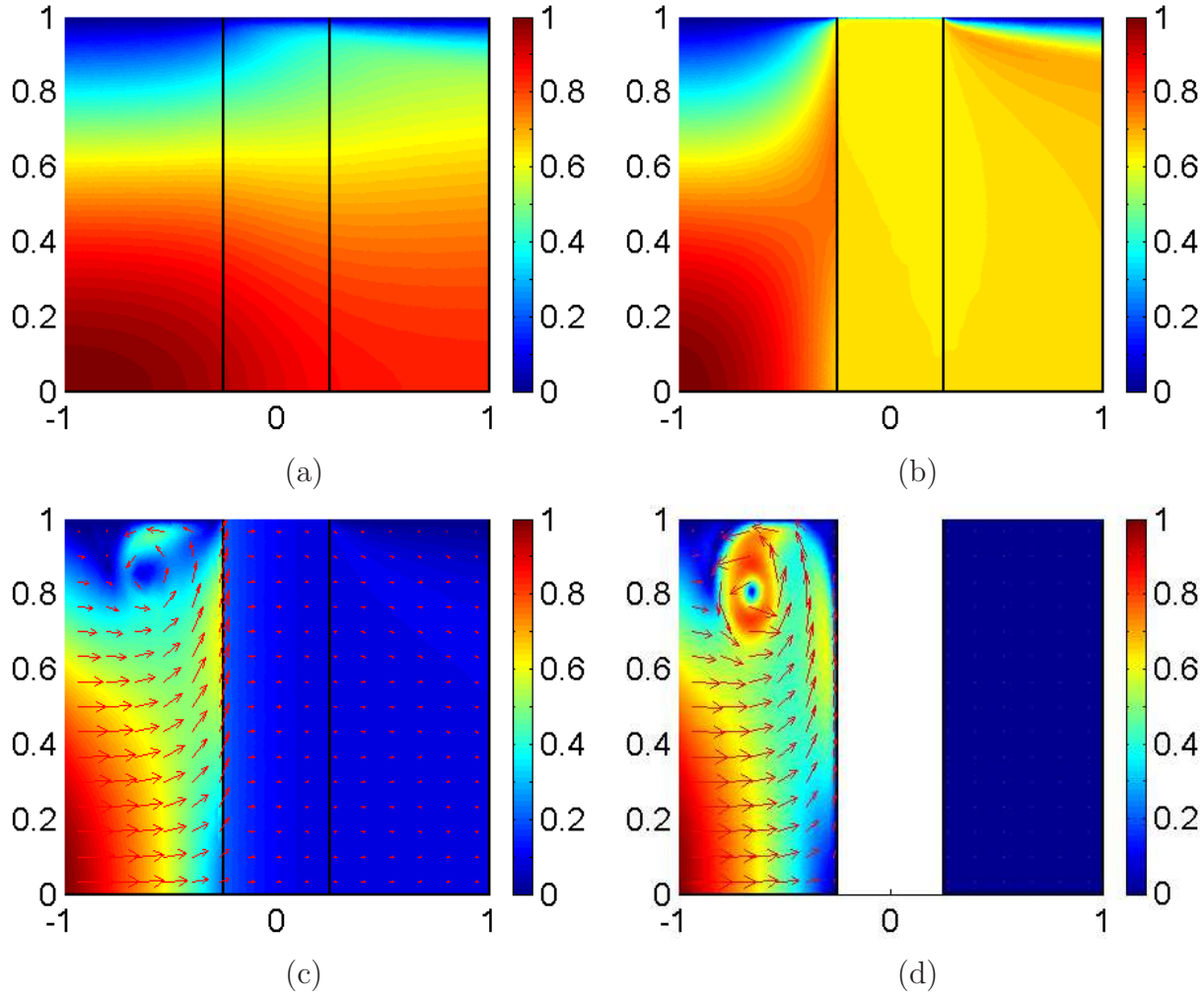


Figure 6. Compressible flow through the porous wall: (a) $\alpha = 1.0$, (b) $\alpha = 1.0 \cdot 10^2$, (c) $\alpha = 1.0 \cdot 10^4$, (d) $\alpha = +\infty$. **Compare with Fig. 5.**

system the variational formulation is as follows. Given $\rho \in \mathcal{H}$, find the solution to the following minimization problem:

$$\min_{\boldsymbol{v} \in \mathcal{U}} \left\{ \mathcal{J}^F(\rho, \boldsymbol{v}) + (2\delta)^{-1} \int_\Omega (\operatorname{div} \boldsymbol{v})^2 \right\}. \tag{5}$$

We note that in the case of homogeneous boundary conditions we have $\mathcal{U} = H_0^1(\Omega)$.

Remark 10. Since the condition $\operatorname{div} \boldsymbol{u} = 0$ is violated, we should replace the term $\int_\Omega |\boldsymbol{\nabla u}|^2$ in the definition of $\mathcal{J}^{\mathcal{S}}$ with $\int_\Omega |E(\boldsymbol{u})|^2$, where $E(\boldsymbol{u}) = (\boldsymbol{\nabla u} + \boldsymbol{\nabla u}^t)/2$ is the linearized rate of strain tensor (see Section 4.3 in Ref. 22). However, both quadratic forms give rise to equivalent norms on $H_0^1(\Omega)$ and thus do not affect our theoretical developments in any way. Therefore, we choose to keep the definition of $\mathcal{J}^{\mathcal{S}}$ for notational simplicity.

In fact, one can go one step further and replace the term $\int_\Omega |\boldsymbol{\nabla u}|^2$ with $\int_\Omega \mathcal{P}(|E(\boldsymbol{u})|)$, where $\mathcal{P}$ is a positive convex function verifying certain growth assumptions, thus including non-Newtonian flows into the discussion (see Chapters 3 and 4 in Ref. 25). For some functionals this will not affect the discussion, while for others (e.g., Prandtl-Eyring fluids) we must reconsider the very basic problem statements [such as the Eq. (5)]. Therefore, in this paper we consider Newtonian fluids only (that is, the case $\mathcal{P}(x) = x^2$) and discuss possible extensions in Section VIII.

Proposition 11. *For every design $\rho \in \mathcal{H}$ the optimization problem (5) has a unique solution $\boldsymbol{v} \in H^1(\Omega)$ whenever its objective functional is proper w.r.t. $\mathcal{U}$, in particular if $\mathcal{U} = H^1_0(\Omega)$.*

Now we are ready to state the main theorem of this section, which establishes the continuity of the design-to-flow mapping in the case of Stokes flow with homogeneous boundary conditions.

Theorem 12. *Consider a sequence of designs $\{\rho_k\} \subset \mathcal{H}$ and the corresponding sequence of flows $\{\boldsymbol{u}_k\} \subset H^1_0(\Omega)$, $k = 1, 2, \ldots$ (i.e., $\boldsymbol{u}_k$ solves the problem (5) for ρ_k). Assume that $\rho_k \to \rho_0$, strongly in $L^1(\Omega + B_R)$, and $\boldsymbol{u}_k \rightharpoonup \boldsymbol{u}_0$, weakly in $H^1_0(\Omega)$. Then, $\boldsymbol{u}_0$ is the flow corresponding to the limiting design ρ_0.*

Remark 13. Theorem 12 shows the epi-convergence of the objective functionals of the ρ-parametric optimization problem (5) as the parameters strongly converge in $L^1(\Omega+B_R)$.

Remark 14. We use strong convergence on the space of designs in order to guarantee the Lipschitz continuity of the family of walls $\{\, \boldsymbol{x} \in \Omega \mid (F * \rho_k)(\boldsymbol{x}) = 0 \,\}$, parametrized by $k \in \mathbb{N}$, which is a stronger property than upper-semicontinuity (cf. Proposition 5). We need Lipschitz continuity to prove Theorem 12.

In the case of non-homogeneous boundary conditions, the result is essentially the same provided we can keep the walls away from the regions of the boundary where injection/suction of the fluid is performed; see Subsection B of Section III and Example 3 for motivations.

Theorem 15. *Consider a sequence of designs $\{\rho_k\} \subset \mathcal{H}$ and the corresponding sequence of flows $\{\boldsymbol{u}_k\} \subset \mathcal{U}$, $k = 1, 2, \ldots$ (i.e., $\boldsymbol{u}_k$ solves the problem (5) for ρ_k). Assume that $\rho_k \to \rho_0$, strongly in $L^1(\Omega + B_R)$, and $\boldsymbol{u}_k \rightharpoonup \boldsymbol{u}_0$, weakly in $H^1(\Omega)$. Further assume that for some positive constants ε, τ it holds that*

$$\inf\{\, (F * \rho_k)(\boldsymbol{x}) \mid k \in \mathbb{N}, \boldsymbol{x} \in \Omega \cap (\operatorname{supp} \boldsymbol{g} + B_\varepsilon) \,\} \geq \tau. \tag{6}$$

Then, $\boldsymbol{u}_0$ is the flow, corresponding to the limiting design ρ_0 (i.e., $\boldsymbol{u}_0$ solves the problem (5) for ρ_0).

Remark 16. We note that the condition (6) is automatically verified for Stokes problems with homogeneous boundary conditions, because the infimum is taken over the empty set in this case ($\operatorname{supp} \boldsymbol{g} = \emptyset$).

B. Navier–Stokes flows

In the case of the Navier–Stokes equations things get much more complicated, because we do not seek a minimizer of some functional anymore, and we cannot apply epi-convergence results directly. Nevertheless, we can utilize them to show the closedness of the design-to-flow mappings even in the Navier–Stokes case.

We introduce a general fixed-point framework related to the optimization problem (5), and then show (at least for the case of homogeneous boundary conditions) that the slightly compressible Navier–Stokes equations can be considered in this framework.

Let $A(\boldsymbol{u}, \boldsymbol{v}) : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ be a weakly continuous functional, and consider the problem of finding a fixed point of the point-to-set mapping $T_\rho : \mathcal{U} \rightrightarrows \mathcal{U}$ defined for $\rho \in \mathcal{H}$ as

$$T_\rho(\boldsymbol{u}) = \operatorname*{argmin}_{\boldsymbol{v} \in \mathcal{U}} \left\{ \mathcal{J}^F(\rho, \boldsymbol{v}) + (2\delta)^{-1} \int_\Omega (\operatorname{div} \boldsymbol{v})^2 + A(\boldsymbol{u}, \boldsymbol{v}) \right\}. \tag{7}$$

Theorem 17. *Consider a sequence of designs* $\{\rho_k\} \subset \mathcal{H}$ *and the corresponding sequence of fixed points* $\{\boldsymbol{u}_k\} \subset \mathcal{U}$, $k = 1, 2, \ldots$ *(i.e.,* $\boldsymbol{u}_k \in T_{\rho_k}(\boldsymbol{u}_k)$ *for* T_{ρ_k} *defined by Eq. (7)). Assume that* $\rho_k \to \rho_0$, *strongly in* $L^1(\Omega+B_R)$, $\boldsymbol{u}_k \rightharpoonup \boldsymbol{u}_0$, *weakly in* $H^1(\Omega)$, *and* $T(\boldsymbol{u}_0) \neq \emptyset$. *Further assume that for some positive constants* ε, τ *the condition (6) is satisfied. Then,* $\boldsymbol{u}_0 \in T_{\rho_0}(\boldsymbol{u}_0)$.

Remark 18. In fact, weak continuity of $A(\boldsymbol{u}, \boldsymbol{v})$ is unnecessarily strong requirement. We can prove Theorem 17 under the following weaker assumptions on A:

(*i*) $A(\boldsymbol{u}, \boldsymbol{u}) \leq \liminf_{k\to\infty} A(\boldsymbol{u}_k, \boldsymbol{u}_k)$ whenever $\boldsymbol{u}_k \rightharpoonup \boldsymbol{u}$, weakly in $\mathcal{U}$; and

(*ii*) $A(\boldsymbol{u}, \boldsymbol{v}) \geq \limsup_{k\to\infty} A(\boldsymbol{u}_k, \boldsymbol{v}_k)$ whenever $\boldsymbol{u}_k \rightharpoonup \boldsymbol{u}$, weakly in $\mathcal{U}$, and $\boldsymbol{v}_k \to \boldsymbol{v}$, strongly in $\mathcal{U}$.

As an example application of Theorem 17, we consider a particular penalty formulation of the incompressible Navier–Stokes equations with homogeneous boundary conditions studied in Ref. 26. A more general treatment is of course possible, including inhomogeneous boundary conditions and variants of slightly compressible Navier–Stokes equations; the main difference is in the number of technical details to be covered.

To put the penalty formulation considered in Ref. 26 (of course, without the control term α) into the fixed-point framework we define

$$A(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega (\boldsymbol{u} \cdot \boldsymbol{\nabla}\boldsymbol{u}) \cdot \boldsymbol{v} + 2^{-1} \int_\Omega (\boldsymbol{u} \cdot \boldsymbol{v}) \operatorname{div} \boldsymbol{u}. \tag{8}$$

We note that the last integral adds an additional stability to the penalty algorithm[26] and identically equals zero in the incompressible case; we can thus expect that the effects of its presence can be almost neglected in the slightly compressible case. Owing to Lemma 2.7 in Ref. 26, the functional A defined in Eq. (8) is weakly continuous on $H^1_0(\Omega) \times H^1_0(\Omega)$, and in order to apply Theorem 17 it remains to establish an analogue of Proposition 11.

Proposition 19. *With* $\mathcal{U} = H^1_0(\Omega)$ *and* A *defined in Eq. (8), the fixed-point problem associated with the operator* $T_\rho(\boldsymbol{u})$ *given in Eq. (7) admits solutions for every* $\rho \in \mathcal{H}$.

Remark 20. While the mapping $(\rho, \boldsymbol{u}) \to T_\rho(\boldsymbol{u})$ is in many cases single-valued for every pair $(\rho, \boldsymbol{u})$, there might be more than one solution to the fixed point problem associated with this operator. In other words, we do not assume that the compressible Navier-Stokes system admits a unique solution.

Remark 21. We can use another popular weak formulation of slightly compressible Navier–Stokes equations,[27] identifying

$$A(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{2}\left(\int_\Omega (\boldsymbol{u} \cdot \boldsymbol{\nabla}\boldsymbol{u}) \cdot \boldsymbol{v} - \int_\Omega (\boldsymbol{u} \cdot \boldsymbol{\nabla}\boldsymbol{v}) \cdot \boldsymbol{u} \right).$$

Our results hold even in this case without any changes.

Remark 22. Of course, the fixed-point framework is not bounded to Navier–Stokes equations. For example, putting, for some $\boldsymbol{u}_0 \in \mathbb{R}^d$,

$$A(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega (\boldsymbol{u}_0 \cdot \boldsymbol{\nabla}\boldsymbol{u}) \cdot \boldsymbol{v},$$

we can show continuity results for Oseen flows. This type of flow is probably not very interesting in bounded domains Ω, but illustrates the possible uses of the fixed-point formulation. Finally, we note that setting $A \equiv 0$ we recover the original Stokes problem.

VI. Existence of optimal solutions

A. Ensuring strong convergence of designs and condition (6)

The results established in Section V all require strong convergence of designs in $L^1(\Omega + B_R)$. In order to guarantee convergence we need to embed our controls $\mathcal{H}$ into some space that is more regular than $L^\infty(\mathbb{R}^d) \cap L^1(\mathbb{R}^d)$. The most appropriate choice, in our opinion, is the space $SBV(\mathbb{R}^d)$, which is typically used for perimeter constrained topology optimization (see p. 31 in Ref. 3 and references therein). Other choices are possible, including $W^{1,1}(\mathbb{R}^d) \cap L^\infty(\mathbb{R}^d)$ (that is, imposing "slope constraints" on the design space[28]). Bounds on the perimeter, or slope, may be introduced into the problem directly as constraints, or added as penalties to the objective function.

Regardless of the particular method used, we get the required property: $\rho_k \rightharpoonup \rho$, weakly in $\mathcal{H}$, implies $\rho_k \to \rho$, strongly in $L^1(\Omega + B_R)$, allowing us to establish the closedness of the design-to-flow mappings.

As for the condition (6), it can be easily verified if we require in addition that every design $\rho \in \mathcal{H}$, satisfying the bounds $0 \le \rho \le 1$, a.e. on $\mathbb{R}^d$, also satisfies $\rho \ge \tau$, a.e. on $\operatorname{supp} \boldsymbol{g} + B_{R+\varepsilon}$, for some positive constants ε, τ.

B. An abstract flow topology optimization problem

Now we are ready to formally discuss the well-posedness of an abstract flow topology optimization problem:

$$\min_{(\rho,\boldsymbol{u})} \mathcal{F}(\rho, \boldsymbol{u}), \qquad \text{s.t.} \begin{cases} (\rho, \boldsymbol{u}) \in \mathcal{Z}, \\ \boldsymbol{u} \in T_\rho(\boldsymbol{u}). \end{cases} \tag{9}$$

The previous results imply the following theorem.

Theorem 23. *Let $\mathcal{Z}$ be a nonempty weakly compact subset of $\mathcal{H}\times\mathcal{U} \subset SBV(\mathbb{R}^d)\times H^1(\Omega)$, and let for all $\rho \in \mathcal{H}$ the assumption (6) be verified (see the discussion in Subsection A). We also assume that A [which defines the mapping T_ρ via Eq. (7)] enjoys the conditions of Remark 18, and that for every $\rho \in \mathcal{H}$ the fixed-point problem associated with T_ρ admits solutions. Finally, let $\mathcal{F} : SBV(\mathbb{R}^d) \times H^1(\Omega) \to \mathbb{R}$ be weakly lower semi-continuous. Then, there exists at least one optimal solution to the abstract flow topology optimization problem (9).*

Remark 24. If the assumptions of Theorem 23 about the flow model are satisfied, we may set

$$\mathcal{Z} = \{\, (\rho, \boldsymbol{u}) \in \mathcal{Z}_0 \times \mathcal{U} \mid \mathcal{G}(\rho, \boldsymbol{u}) \le C \,\},$$

where $\mathcal{Z}_0$ is a nonempty weakly compact subset of $\mathcal{H} \subset SBV(\mathbb{R}^d)$ verifying condition (6), $\mathcal{G}(\rho, \boldsymbol{u})$ is an arbitrary weakly l.s.c. functional, which is in addition coercive in $\boldsymbol{u}$, uniformly w.r.t. ρ, and $C \in \mathbb{R}$ is an arbitrary constant but such that $\mathcal{Z} \ne \emptyset$.

In particular, we may set $\mathcal{G} = \mathcal{J}$, or $\mathcal{G} = \mathcal{J}^F$ (see Lemma 3.2 in Ref. 9).

At last, we note that assumptions of Theorem 23 about the solvability of the fixed-point problem for every feasible design ρ are verified in many practical situations. For example, we have shown that they are satisfied for Stokes equations (see Proposition 11 and Remark 22) and for Navier–Stokes equations with homogeneous boundary conditions (see Proposition 19).

VII. Computational issues

In this section we briefly discuss two topics that are standard in topology optimization with specialization to flow topology optimization problems. Throughout the section we will use the problem (9) as a model example, and we assume that the assumptions of Theorem 23 are verified without further notice.

A. Approximation with sizing optimization problems

Clearly, no finite element software can be applied to the minimization problem appearing in Eq. (7) if $\alpha(F * \rho)$ is allowed to become arbitrarily large; from the practical point of view the theory of Section V implying the existence of optimal solutions to the problem (9) is pointless, unless we can describe a computational procedure capable of finding approximations of these optimal solutions. In fact, once we have proved Theorem 17 the latter goal can be easily accomplished. For arbitrary $\varepsilon > 0$, consider the set $\mathcal{Z}_\varepsilon = \{\, (\rho, \boldsymbol{u}) \in \mathcal{Z} \mid \rho \geq \varepsilon, \text{a.e.} \,\}$, i.e., only designs with porosity uniformly bounded away from zero are allowed, implying in particular the uniform bound $\alpha(F * \rho) \leq \varepsilon^{-1} - 1$, for every $(\rho, \boldsymbol{u}) \in \mathcal{Z}_\varepsilon$.

Then, the following easy statement holds.

Proposition 25. *Assume that the sequence $\{\mathcal{Z}_\varepsilon\}$ is lower-semicontinuous in Painlevé-Kuratowski sense (topology in $\mathcal{H} \times \mathcal{U}$ being the strong one), namely*

$$\liminf_{\varepsilon \to +0} \mathcal{Z}_\varepsilon = \mathcal{Z}, \tag{10}$$

(in particular, $\mathcal{Z}_\varepsilon \neq \emptyset$ for all small $\varepsilon > 0$). Let further, for every small $\varepsilon > 0$, $(\rho_\varepsilon, \boldsymbol{u}_\varepsilon)$ denote a globally optimal solution of an approximating problem, obtained from the problem (9) substituting $\mathcal{Z}_\varepsilon$ in place of $\mathcal{Z}$. Then, an arbitrary limit point of $\{(\rho_\varepsilon, \boldsymbol{u}_\varepsilon)\}$ (and there is at least one) is a globally optimal solution of the limiting problem (9).

The assumption (10) is probably easier to check in every particular case rather than to develop a general sufficient condition implying it; we only mention that for typical constraints in topology optimization, such as constraints on volume and on the perimeter, it is easily verified.

In general, there is a substantial amount of literature on the topic of approximation of topology optimization problems using sizing ones. (See the bibliographical notes [16] in Ref. 3 for a survey of the situation in the topology optimization of linearly elastic materials; also see Section 6 in Ref. 9 for results on incompressible stokesian flows, and Appendix A.2 in Ref. 10 for a similar problem arising in the design of flow networks.) Cases of interest in such literature are when some of the underlying assumptions of Proposition 25 are violated, such as the compactness of $\mathcal{Z}$ or $\mathcal{Z}_\varepsilon$, or the assumption (10); in some particular situations it is nevertheless possible to prove statements similar to Proposition 25. We do not try to generalize our result in this direction, because computationally the problem (9) is already extremely demanding for realistic flows, and complicated constraints violating the assumption (10) are hardly necessary in practical situations.

B. Control of intermediate densities

Starting with the problem of distributing the solid material inside a control volume Ω so as to minimize some objective functional dependent on the flow, we expect an optimal design of the type $\rho = \chi_A$, where $A \subset \Omega$ is a flow region ("black–white" designs).

Usually, this is a very naïve expectation (see Section 1.3.1 in Ref. 3); however, there are some exceptions, such as the minimum-power design of domains for Stokes flows,[8,9] or the design of one-dimensional wave-guides for stopping wave propagation.[29]

However, if we use a filter, it is simply impossible to obtain optimal distributions of material assuming *only* values zero or one (not counting the trivial designs $F * \rho \equiv 0$ and $F * \rho \equiv 1$), because $F * \chi_A$ is a continuous function, and the "edges" ∂A will be "smoothed out" by the filter. One possible way to reduce the amount of porous material in the final optimal design $F * \rho$ is to use a filter of a smaller radius. This may or may not work as expected — since the control problem (9) is non-convex, the optimal designs may change significantly as we vary the radius only slightly.

Another possibility is to add a penalty term $\mu \mathcal{J}^{\mathcal{D}}(F * \rho, \boldsymbol{u})$, for some positive μ, requiring that the power dissipation due to the flow through the porous part of the domain should be relatively small (see Section 5 in Ref. 9). We must warn that increasing penalty μ might lead to unexpected results, because as we have already mentioned, the presence of the filter *requires* the presence of porous regions in the domain (except for trivial cases), thus the sequence of designs may converge to either one of those trivial designs, or $\mu \mathcal{J}^{\mathcal{D}}(F * \rho, \boldsymbol{u})$ may grow indefinitely. Therefore, suitable values of μ should be obtained in each case experimentally.

At last, various restriction or regularization techniques that are designed to control the amount of "microstructural material" in topology optimization of linearly elastic structures may be used for similar purposes in our case. We already mentioned the regularized intermediate density control method;[19] other possible choices may be found in bibliographical notes [8] in Ref. 3.

VIII. Conclusions and further research

We have considered the topology optimization of fluid domains in a rather abstract setting, and established the closedness of design-to-flow mappings for a general family of slightly compressible fluids, whose behavior is characterized by the fixed-point formulation associated with the operator defined in Eq. (7). We used the notion of epi-convergence of optimization problems as a main analytical tool that allows us to treat very ill-behaving functionals, which arise due to the fact that we allow completely impenetrable walls to appear in the design domain.

It is of course of great engineering interest to perform numerical experiments with topology optimization of slightly compressible fluids for various objective functionals, theoretical foundations for which are established in this paper. Provided a stable solver of the underlying flow problem is available, it should not be a difficult task to combine it with the optimization code; in the end, the ease of integration with FEM software is one of the main reasons why topology optimization techniques are widely accepted and still gain popularity in many fields of physics and engineering.[3] In fact, one such successful attempt of integrating topology optimization with Femlab is done for incompressible Navier–Stokes fluids.[12] Unfortunately, at the time of writing this code was not available to the author. We hope to be able to perform numerical computations in the near future.

The motivation for relaxing the incompressibility requirement is found in Subsection B of Section III; however, if one is not convinced, and for the sake of completeness it would be interesting to prove Theorem 12 for divergence-free functions, from which the rest of the theory should follow for incompressible fluids as well.

The method we used is of course not bound to Newtonian fluids. It seems that our results should hold for many common non-Newtonian fluids, including power-law, Bingham,

and Powell-Eyring models (see Chapter 3 in Ref. 25), without any major modifications (cf. Remark 10). Additional work is obviously needed for fluids of Prandtl-Eyring type (see Chapter 4 in Ref. 25); we however feel that the special treatment this (mathematically) exotic type of fluids deserves lies well outside the scope of this paper.

At last, but not the least, we feel it is important to establish the existence of solutions, or construct a disproving counter-example, for the "original" problem of power minimization for incompressible Navier–Stokes fluids without the use of filtered designs. While we have shown that this problem looks ill-posed and is probably unsuitable for practical numerical computations, knowing whether optimal solutions exist would greatly contribute to the deeper understanding of Navier–Stokes flows and affect the further development in the area of topology optimization of fluids.

Acknowledgments

This research is supported by the Swedish Research Council (grant 621-2002-5780), which is gratefully acknowledged.

References

[1]Gunzburger, M. D., *Perspectives in flow control and optimization*, Advances in Design and Control, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2003.

[2]Mohammadi, B. and Pironneau, O., *Applied shape optimization for fluids*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2001.

[3]Bendsøe, M. P. and Sigmund, O., *Topology Optimization: Theory, Methods, and Applications*, Springer-Verlag, Berlin, 2003.

[4]Feireisl, E., "Shape optimization in viscous compressible fluids," *Appl. Math. Optim.*, Vol. 47, No. 1, 2003, pp. 59–78.

[5]Ton, B. A., "Optimal shape control problem for the Navier-Stokes equations," *SIAM J. Control Optim.*, Vol. 41, No. 6, 2003, pp. 1733–1747.

[6]Gunzburger, M. D., Kim, H., and Manservisi, S., "On a shape control problem for the stationary Navier-Stokes equations," *M2AN Math. Model. Numer. Anal.*, Vol. 34, No. 6, 2000, pp. 1233–1258.

[7]Gunzburger, M. D. and Kim, H., "Existence of an optimal solution of a shape control problem for the stationary Navier-Stokes equations," *SIAM J. Control Optim.*, Vol. 36, No. 3, 1998, pp. 895–909.

[8]Borrvall, T. and Petersson, J., "Topology optimization of fluids in Stokes flow," *Internat. J. Numer. Methods Fluids*, Vol. 41, No. 1, 2003, pp. 77–107.

[9]Evgrafov, A., "On the limits of porous materials in the topology optimization of Stokes flows," Blå serien, Department of Mathematics, Chalmers University of Technology, Gothenburg, Sweden, 2003, Submitted for publication.

[10]Klarbring, A., Petersson, J., Torstenfelt, B., and Karlsson, M., "Topology optimization of flow networks," *Comput. Methods Appl. Mech. Engrg.*, Vol. 192, No. 35-36, 2003, pp. 3909–3932.

[11]Darrigol, O., "Between hydrodynamics and elasticity theory: the first five births of the Navier-Stokes equation," *Arch. Hist. Exact Sci.*, Vol. 56, No. 2, 2002, pp. 95–150.

[12]Gersborg-Hansen, A., *Topology optimization of incompressible Newtonian flows at moderate Reinolds numbers*, Master's thesis, Department of Mechanical Engineering, Technical University of Denmark, December 2003.

[13]Allaire, G., "Homogenization of the Navier-Stokes equations in open sets perforated with tiny holes. I. Abstract framework, a volume distribution of holes," *Arch. Rational Mech. Anal.*, Vol. 113, No. 3, 1990, pp. 209–259.

[14]Sigmund, O., "On the design of compliant mechanisms using topology optimization," *Mech. Struct. Mach.*, Vol. 25, No. 4, 1997, pp. 493–524.

[15]Sigmund, O. and Petersson, J., "Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima," *Struct. Multidisc. Optim.*, Vol. 16, No. 1, 1998, pp. 68–75.

[16]Bourdin, B., "Filters in topology optimization," *Internat. J. Numer. Methods Engrg.*, Vol. 50, No. 9, 2001, pp. 2143–2158.

[17]Bruns, T. E. and Tortorelli, D. A., "Topology optimization of non-linear elastic structures and compliant mechanisms," *Comput. Methods Appl. Mech. Engrg.*, Vol. 190, No. 26–27, 2001, pp. 3443–3459.

[18]Allaire, G., "Homogenization of the Navier-Stokes equations in open sets perforated with tiny holes. II. Noncritical sizes of the holes for a volume distribution and a surface distribution of holes," *Arch. Rational Mech. Anal.*, Vol. 113, No. 3, 1990, pp. 261–298.

[19]Borrvall, T. and Petersson, J., "Topology optimization using regularized intermediate density control," *Comput. Methods Appl. Mech. Engrg.*, Vol. 190, No. 37–38, 2001, pp. 4911–4928.

[20]Hornung, U., editor, *Homogenization and porous media*, Vol. 6 of *Interdisciplinary Applied Mathematics*, Springer-Verlag, New York, 1997.

[21]Sokołowski, J. and Zolésio, J.-P., *Introduction to shape optimization*, Vol. 16 of *Springer Series in Computational Mathematics*, Springer-Verlag, Berlin, 1992.

[22]Gunzburger, M. D., *Finite element methods for viscous incompressible flows*, Computer Science and Scientific Computing, Academic Press Inc., Boston, MA, 1989.

[23]Temam, R., *Navier-Stokes equations*, AMS Chelsea Publishing, Providence, RI, 2001, Reprint of the 1984 edition.

[24]Heinrich, J. C. and Vionnet, C. A., "The penalty method for the Navier-Stokes equations," *Arch. Comput. Methods Engrg.*, Vol. 2, No. 2, 1995, pp. 51–65.

[25]Fuchs, M. and Seregin, G., *Variational methods for problems from plasticity theory and for generalized Newtonian fluids*, Vol. 1749 of *Lecture Notes in Mathematics*, Springer-Verlag, Berlin, 2000.

[26]Carey, G. F. and Krishnan, R., "Penalty finite element method for the Navier-Stokes equations," *Comput. Methods Appl. Mech. Engrg.*, Vol. 42, No. 2, 1984, pp. 183–224.

[27]Lin, S. Y., Chin, Y. S., and Wu, T. M., "A modified penalty method for Stokes equations and its applications to Navier-Stokes equations," *SIAM J. Sci. Comput.*, Vol. 16, No. 1, 1995, pp. 1–19.

[28]Petersson, J. and Sigmund, O., "Slope constrained topology optimization," *Internat. J. Numer. Methods Engrg.*, Vol. 41, No. 8, 1998, pp. 1417–1434.

[29]Bellido, J. C., "Existence of classical solutions for a one-dimensional optimal design problem in wave propagation," 2003, Preprint, Mathematical Institute, University of Oxford, Oxford, UK. Submitted for publication.

A Generating Set Search Method Exploiting Curvature and Sparsity*

Lennart Frimannslund† Trond Steihaug‡

Abstract

Generating Set Search methods are one of the few alternatives for optimising high fidelity functions with numerical noise. These methods are usually only efficient when the number of variables is relatively small. This paper presents a modification to an existing Generating Set Search method, which makes it aware of the sparsity structure of the Hessian. The aim is to enable the efficient optimisation of functions with a relatively large number of variables. Numerical results show a decrease in the number of function evaluation it takes to reach the optimal solution, sometimes by significant margins, on noisy as well as smooth problems, for a modest as well as a relatively large number of variables.

Keywords: Nonlinear programming, derivative–free optimization, pattern search, generating set search, sparsity.

1 Introduction

We consider the unconstrained optimisation problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$. Suppose that f is only available as

$$\widetilde{f}(x) = f(x) + \epsilon, \tag{2}$$

where the error term ϵ is either stochastic or numerical in nature. By numerical noise we mean the noise which can arise from, for instance, the discretisation involved if evaluating

*This work was supported by the Norwegian Research Council.

†Department of Informatics, University of Bergen, Box 7800, N-5020 Bergen, Norway. E-mail: lennart.frimannslund@ii.uib.no

‡Department of Informatics, University of Bergen. E-mail: trond.steihaug@ii.uib.no

f requires computing an integral, solving a differential equation or any other subproblem which is solved inexactly. The same input will always give the same output, but the function will not be smooth. An example of such a function occurs in [1], where the objective function contains an integral. The truncation error stemming from the computation of the integral makes the function look like the one in figure 1. There is an underlying smooth function, but it is obscured by noise. On such methods derivative-based methods can easily run into trouble, since finite difference-based derivatives may be very inaccurate and automatic differentiation often is unhelpful as well. Generating Set Search (GSS) Methods are a good alternative in this case. GSS methods are comprehensively reviewed in [12]. Although usually easy to implement, GSS methods in their most basic form often converge slowly. Modifications to speed up convergence were suggested as early as in 1960 by Rosenbrock [17]. Two recent approaches using curvature information have been suggested [2, 7]. The main modification to basic GSS in these papers is that the search directions the methods consider are dynamic. The introduction of a dynamic search basis is shown to significantly reduce the number of function evaluations required to reach the optimiser, in most cases.

Apart from slow convergence, GSS methods are often unsuitable for problems where the number of variables n is large. In [16], one proposes a method effective for the optimisation of smooth functions which can be decomposed into *element functions*. Let $\chi_k \subseteq \{1, 2, \ldots, n\}$, $k = 1, \ldots, n$ and let $|\chi_k|$ be the cardinality of the set χ_k. Let $f_k : \mathbb{R}^{|\chi_k|} \mapsto \mathbb{R}$, $k = 1, \ldots, n$, where χ_k are the indices of x on which f_k depend. If f is of the form

$$f(x) = \sum_{k=1}^{n} f_k(x), \tag{3}$$

then f is said to be *partially separable*, or *totally separable* depending on the cardinality of the sets χ_k. Separability of f is closely related to the sparsity structure of the derivatives, but we make the distinction because separability structure is defined even if the function is not differentiable. Theory on separability of functions can be found in [11].

Given a totally separable function one can obtain the value of f at as many as $3^n - 1$ points at the cost of only 2 f-evaluations, as long as the points in question are aligned with the coordinate axes. The optimisation algorithm in [16] exploits this fact to solve smooth problems of the form (1) with f of the form (3) for up to more than 5000 variables. We wish to exploit separability of f, on noisy functions.

In [7] an algorithm which solves (1) where the function is of the form (2) using average curvature information to speed up convergence was developed. However, as n grows, the algorithm becomes increasingly unable to exploit this information. In this paper we present an extension to the algorithm of [7], which utilises the sparsity pattern of the Hessian of f in (2). Although noise can potentially eliminate any sparsity pattern from $\nabla^2 f$ in $\nabla^2 \widetilde{f}$, a priori knowledge about $\nabla^2 f$ through knowledge about the separability structure (3) or known Hessian sparsity structure is assumed to be valid for $\nabla^2 \widetilde{f}$ as well.

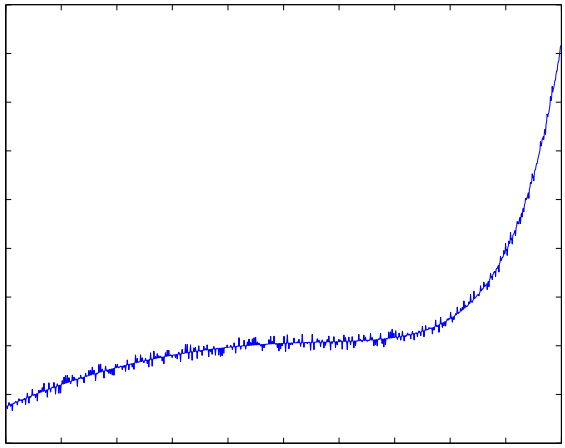


Figure 1: $e^x - x^2$ with noise.

This paper focuses on unconstrained optimisation, but extensions toward constrained optimisation discussed in [4, 13, 14] are applicable.

2 Generating Set Search

GSS methods are a class of methods which search along the vectors of a *generating set* or *positive basis*. A generating set consists of vectors $v_i,\ i = 1, \ldots, r$ such that for any $x \in \mathbb{R}^n$,

$$x = \sum_{i=1}^{r} c_i v_i,\ c_i \geq 0,\ i = 1, \ldots, r.$$

In words, the vectors in the set *positively span* $\mathbb{R}^n$. It is shown in [3] that to positively span $\mathbb{R}^n$, $n + 1 \leq r \leq 2n$, depending on the vectors. The positive and negative of the Cartesian coordinate vectors, say $e_i,\ i = 1, \ldots, n$ are an example of a generating set with $2n$ vectors. These methods are also known as *pattern search*, the name Generating Set Search was coined in [12].

Let the set of search directions $\mathcal{D}$ be defined as

$$\mathcal{D} = \bigcup_{i=1}^{r} \{p_i\}.$$

Associate with each p_i a step length δ_i. Then, a pseudo code for a method we will call *Compass Search* is:

Compass Search

Given $x,\ \delta_{tol},\ \alpha \geq 1 > \beta > 0,$

Repeat until convergence,

For each $p_i \in \mathcal{D}$,

If $f(x + \delta_i p_i) < f(x)$,

$x \leftarrow x + \delta_i p_i$

$\delta_i \leftarrow \alpha \delta_i$

else,

$\delta_i \leftarrow \beta \delta_i$

end.

end.

end.

α and β need not be constant throughout. We will call one run of the repeat-loop a *sweep*. For this and other GSS methods one can expect linear convergence, see [12] and the references therein.

Rosenbrock's method [17] is based on Compass Search with $2n$ search directions. It regularly rotates the search vectors in $\mathcal{D}$ by aligning the principal search direction to an average gradient and generates $(n-1)$ additional directions through the Gram-Schmidt process. It uses the positive and negative of the resulting vectors as its new search directions.

2.1 GSS Methods Using Curvature Information

We look at two different methods employing curvature information.

The Method of Coope and Price This method for unconstrained optimisation of smooth functions, is described fully in [2]. It minimises the function on successively finer grids which are defined by the search directions $v_i, i = 1, \ldots, n$ and the step lengths associated with each direction. The method searches along both the positive and negative of these directions, and hence has $2n$ search directions. In the process of searching along the current direction, say, v_i, the method obtains the function values at three points along this line. From these three points it creates an interpolating quadratic function. The step length δ_i corresponding to v_i is then based on the distance from the current iterate to the minimiser of the interpolating function.

Using the parallel subspace theorem (see, e.g. theorem 4.2.1 of [6]) the method generates conjugate search directions, one direction at a time from the n initially non-conjugate search directions. Once a conjugate direction has been found, the algorithm deletes a non-conjugate direction, to maintain the number of search directions. The generated conjugate directions are stored in a matrix V_c, which becomes an indirect approximation to $(\nabla^2 f)^{-1}$ once n conjugate directions have been found, by the relation

$$V_c V_c^T \approx (\nabla^2 f)^{-1}.$$

The method is able to perform a finite difference Newton step from time to time. Once the entire inverse Hessian approximation is in place, the algorithm starts building up a new approximation. The algorithm terminates exactly on quadratic functions.

A Method Exploiting Average Curvature Information This method is described fully in [7]. Let the search basis $\mathcal{D}$ consist of the positive and negative of the column vectors of the orthogonal matrix

$$Q = \left[\begin{array}{cccc} q_1 & q_2 & \cdots & q_n \end{array} \right],$$

where q_i is column i. By adaptively shuffling the order of the directions in $\mathcal{D}$ once per sweep, the algorithm is able to gather average curvature information from the history of function evaluations. The algorithm builds up what in [7] is called a *curvature information matrix*, C_Q, one element at the time, by the formula

$$(C_Q)_{ij} = \frac{f(x^{ij} + \delta_i q_i + \delta_j q_j) - f(x^{ij} + \delta_i q_i) - f(x^{ij} + \delta_j q_j) + f(x^{ij})}{\delta_i \delta_j}. \tag{4}$$

where δ_i and δ_j are the step lengths along the search directions q_i and q_j respectively, at any given time. The point x^{ij} is usually different for each $(C_Q)_{ij}$. C_Q is required to be symmetric, so only the lower triangle of C_Q is computed. The expression (4) equals a directional second derivative,

$$(C_Q)_{ij} = q_i^T \nabla^2 f(\widetilde{x}^{ij}) q_j \tag{5}$$

for some $\widetilde{x}^{ij}$ in the rectangle with the four points $x^{ij} + \delta_i q_i + \delta_j q_j$, $x^{ij} + \delta_i q_i$, $x^{ij} + \delta_j q_j$ and x^{ij} as corner points. (See e.g. lemma 3.5 in [5].) If the step lengths are sufficiently large then average curvature information is obtained, thus smoothing out the effects of noise. The method is able to obtain $O(n)$ C_Q-elements per sweep, so the entire matrix C_Q consisting of $\frac{n^2+n}{n}$ unique elements is computed in $O(n)$ sweeps. When C_Q is determined, the matrix C, given by the formula

$$C = Q C_Q Q^T, \tag{6}$$

is computed. The positive and negative of the eigenvectors of C are taken as the new search basis, and Q is updated accordingly.

3 A Scheme for Exploiting Sparsity

We now propose an extension to the algorithm of [7]. Assume f is separable. The individual f_k and χ_k define $|\chi_k| \times |\chi_k|$ Hessian structural information, and by assembling all the individual matrices, we have a sparsity structure for the entire Hessian. If sparsity structure is not known a priori, it can be detected by the technique of [10], or it is possible to obtain the information from computational graphs, which are used in Automatic Differentiation (AD). (See, e. g. [9] for more on AD.)

However, sparsity is relative to the coordinate system. C_Q will not be sparse if $Q \neq I$, and neither will the matrix C from (6) be unless the function is quadratic, due to truncation error in (4). Therefore, we impose the restriction that C have the same sparsity structure as the Hessian.

When $\nabla^2 f$ is full, we need to compute $\frac{n^2+n}{2}$ C_Q-elements by (4). If the Hessian is sparse with, for instance, $O(n)$ unique elements, we would like to compute no more elements in C_Q than there are unique elements in the Hessian itself. $O(n)$ elements can be computed in $O(1)$ sweeps.

We do this by writing (6) as the equation

$$Q^T C Q = C_Q, \tag{7}$$

where the unknown is the matrix C. Let D and B be $n \times n$-matrices. The *Kronecker product* $(D \otimes B)$ is an $n^2 \times n^2$-matrix

$$(D \otimes B) = \begin{bmatrix} D_{11}B & \cdots & D_{1n}B \\ \vdots & & \vdots \\ D_{n1}B & \cdots & D_{nn}B \end{bmatrix}. \tag{8}$$

See e.g. [8]. Useful identities are

$$(D \otimes B)^{-1} = (D^{-1} \otimes B^{-1}), \tag{9}$$

and

$$(D \otimes B)^T = (D^T \otimes B^T), \tag{10}$$

Using the Kronecker product, (7) can be rewritten as

$$(Q^T \otimes Q^T)\mathbf{vec}(C) = \mathbf{vec}(C_Q), \tag{11}$$

where **vec** is an operator $\mathbf{vec} : \mathbb{R}^{n \times n} \mapsto \mathbb{R}^{n^2}$ which stacks the entries of a matrix in a vector such that the equivalence between (7) and (11) holds. Denote the columns of the matrix C by $c_i,\ i = 1, \ldots, n$, that is,

$$C = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix}.$$

Then

$$\mathbf{vec}(C) = (\ c_1^T \quad c_2^T \quad \cdots \quad c_n^T\ )^T. \tag{12}$$

If we examine the matrix $(Q^T \otimes Q^T)$ it reads

$$(Q^T \otimes Q^T) = \begin{bmatrix} Q_{11}Q^T & \cdots & Q_{n1}Q^T \\ \vdots & & \vdots \\ Q_{1n}Q^T & \cdots & Q_{nn}Q^T \end{bmatrix}. \tag{13}$$

The first row consists of products involving only the elements of q_1. The second row consists of products involving only the elements of q_1 and q_2. Similarly, each of the remaining rows contain products involving elements of only two q-vectors. Since the **vec**-operator is also applied to C_Q in the right-hand side of (11), the row made up of the vectors q_i and q_j corresponds to the element $(C_Q)_{ij}$ in $\mathbf{vec}(C_Q)$. We now want to reduce the number of variables in (11) based on our knowledge of symmetry and sparsity structure. Since we require C to be symmetric we can, for all $r > s$, add the columns corresponding to C_{sr} to the columns corresponding to C_{rs} and delete the former columns. This means we only consider the elements in the lower triangle of C. Accordingly, we delete all the rows which do not correspond to computation of elements in the lower triangle of C_Q. Furthermore, since C has a certain sparsity structure, we can delete all columns which correspond to elements C_{rs} we know are to be zero.

Having removed the columns corresponding to zero elements, we must also remove the same number of rows. We have some freedom when it comes to which rows are to be removed. We want the resulting coefficient matrix after row removal to be well conditioned. If we were working in a Cartesian coordinate system, then the two vectors used to compute C_{rs} by a difference formula like the one in (4) would be the coordinate vectors e_r and e_s, and any nonsingular submatrix of (13) would be well conditioned. Since we are working in the coordinate system defined by the vectors $q_i,\ i = 1, \ldots, n$, the closest we can get to e_r and e_s are the vectors with their maximum absolute elements in position r and s, that is, vectors q_i and q_j such that

$$\max_k |(q_i)_k| = |(q_i)_r|,$$

and

$$\max_k |(q_j)_k| = |(q_j)_s|.$$

So, for each nonzero C_{rs} we pick the vectors q_i and q_j and keep the corresponding row. Let ρ be the number of unique nonzero elements in the Hessian. Since we want an equation system with ρ equations an unknowns, we need to modify the **vec** to take this into account. Let $\overline{\mathbf{vec}}$ be the operator which stacks the nonzero elements of the lower triangle of a matrix in a vector. Let c_Q signify the ρ-vector of C_Q-entries that we compute. The resulting $\rho \times \rho$ equation system becomes

$$A\overline{\mathbf{vec}}(C) = c_Q, \tag{14}$$

where A is the resulting matrix from modifying $(Q^T \otimes Q^T)$. In our experiments, using the heuristic just described, A was usually very well conditioned.

Since we need to compute ρ c_Q-elements and can compute $O(n)$ elements per sweep, the right-hand side c_Q will be available in $O(\frac{\rho}{n})$ sweeps. Then we solve (14) and construct C with the inverse of the operator $\overline{\mathbf{vec}}$.

3.1 The Relationship between C and the Hessian

In this section we examine the error

$$\|C - \nabla^2 f\|.$$

First we need a technical result. Define

$$c = \overline{\mathbf{vec}}(C),$$

Then we have

$$\|c\| \leq \|C\|_F \leq \sqrt{2}\|c\|. \tag{15}$$

Too see this, suppose that C has n diagonal and γ off-diagonal nonzero elements. We then have

$$\|c\| = \left(\sum_{i=1}^{n+\gamma} c_i^2 \right)^{\frac{1}{2}}, \tag{16}$$

and

$$\|C\|_F = \left(\sum_{\forall(r,s)} C_{rs}^2 \right)^{\frac{1}{2}}. \tag{17}$$

Not counting terms C_{rs}^2 where C_{rs} is known to be zero, the sum in (17) contains $n + 2\gamma$ nonnegative elements. All of the terms in the sum in (16) are present in (17), so clearly $\|c\| \leq \|C\|_F$. As for the second inequality, we have

$$\sqrt{2}\|c\| = \|\sqrt{2}c\| = \left(\sum_{i=1}^{n+\gamma} (\sqrt{2}c_i)^2 \right)^{\frac{1}{2}}. \tag{18}$$

This can be written

$$\left(2\sum_{i=1}^{n+\gamma} c_i^2 \right)^{\frac{1}{2}} = \left(\sum_{i=1}^{n+\gamma} c_i^2 + \sum_{i=1}^{n+\gamma} c_i^2 \right)^{\frac{1}{2}}. \tag{19}$$

The final sum of (19) contains a sum of $2n + 2\gamma$ nonnegative elements. All the $n + 2\gamma$ elements in (17), (still not counting terms C_{rs}^2 where C_{rs} is known to be zero) are present in (19), so the second inequality of (15) holds as well.

Now we can turn our attention to the relationship between C and the Hessian.

Lemma 1 *Let f be twice continuously differentiable. Assume A in (14) is invertible and let c be the solution to (14). Let element l, $l = 1, \ldots, \rho$ of c_Q in (14) be computed by (4) and be equal to $q_i^T \nabla^2 f(\widetilde{x}^l) q_j$ for the appropriate vectors q_i and q_j by (5). Define*

$$N = \bigcup_{l=1}^{\rho} \left\{ \widetilde{x}^l \right\}, \tag{20}$$

and let

$$\delta = \max_{x,y \in N} \|x - y\|, \tag{21}$$

and

$$\mathcal{N} = \left\{ x \in \mathbb{R}^n \,\middle|\, \max_{y \in N} \|x - y\| \leq \delta \right\}. \tag{22}$$

Let f be Lipschitz-continuous in $\mathcal{N}$ with Lipschitz-constant L. Then, the matrix C obtained by applying the inverse of the operator $\overline{\mathbf{vec}}$ on c, satisfies

$$\|C - \nabla^2 f(x)\| \leq \sqrt{2} \rho \kappa(A) L \delta,$$

where $x \in \mathcal{N}$ and $\kappa(A)$ is the condition number of A.

Proof. Let $h_l = \overline{\mathbf{vec}}(\nabla^2 f(\widetilde{x}^l)),\ l = 1, \ldots, \rho$. The Hessian has the same sparsity structure as C, so c_Q can be written

$$c_Q = \begin{bmatrix} (Ah_1)_1 \\ (Ah_2)_2 \\ \vdots \\ (Ah_\rho)_\rho \end{bmatrix},$$

where $(Ah_l)_l$ is the lth element of the vector Ah_l. If we now let E_l be the matrix with 1 in position (l, l) and zero everywhere else, we have

$$c = A^{-1} \sum_{l=1}^{\rho} (E_l A h_l).$$

The Hessian mapping $\nabla^2 f : \mathbb{R}^n \mapsto \mathbb{R}^{n \times n}$ is assumed to be Lipschitz-continuous in $\mathcal{N}$, that is,

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L \|x - y\| \quad \text{for all } x, y \in \mathcal{N}. \tag{23}$$

Let $x \in \mathcal{N}$. Define

$$\overline{\mathbf{vec}}(\nabla^2 f(x)) = h.$$

Then we have

$$c = A^{-1} \sum_{l=1}^{\rho} (E_l A (h + \epsilon_l)),$$

where

$$\epsilon_l = h_l - h.$$

This expands to

$$c = A^{-1}(E_1 + \cdots + E_\rho)Ah + \sum_{l=1}^{\rho} A^{-1}E_l A\epsilon_l.$$

The first part of the expression reduces to just h, since the sum of the E_l becomes the identity matrix. The second term becomes an error term, whose norm is bounded by

$$\|c - h\| = \|\sum_{l=1}^{\rho} A^{-1}E_l A\epsilon_l\| \leq \rho\|A^{-1}\| \left(\max_l \|E_l\|\right) \|A\| \left(\max_l \|\epsilon_l\|\right). \tag{24}$$

All the E_l have unit norm, and the norms $\|A\|$ and $\|A^{-1}\|$ together make up the condition number of the matrix A, $\kappa(A)$. We now need a bound on $\max_l \|\epsilon_l\|$. We have

$$\max_l \|\widetilde{x}^l - x\| \leq \delta,$$

since x and all the $\widetilde{x}^l$ are in $\mathcal{N}$. Thus, by (23):

$$\max_l \|\widetilde{x}^l - x\| \leq \delta \Rightarrow \max_l \|\nabla^2 f(\widetilde{x}^l) - \nabla^2 f(x)\| \leq L\delta.$$

By (15) we have

$$\max_l \|\epsilon_l\| = \max_l \|h - h_l\| \leq \max_l \|\nabla^2 f(\widetilde{x}^l) - \nabla^2 f(x)\|_F \leq L\delta.$$

This turns (24) into

$$\|c - h\| \leq \rho\kappa(A)L\delta,$$

and finally, by (15),

$$\|C - \nabla^2 f(x)\|_F \leq \sqrt{2}\rho\kappa(A)L\delta.$$

□

4 Preliminary Numerical Results

Numerical test were performed on three functions from [15], for various sizes of n. All the functions have a minimum value of zero. The results on smooth functions are listed in table 1. The columns contain, from left to right, the number of variables, the number of unique nonzero elements to be determined ρ, the number of function evaluations performed to reach the solution, the number of C-matrices computed and hence the number of times the positive basis $\mathcal{D}$ is updated, and the final function value obtained, for the method

using sparsity and the method of [7] (marked "regular" in the table), respectively. The convergence criterion used in the experiments on smooth functions was

$$\max_i \delta_i < 10^{-7}.$$

The results on the extended Rosenbrock function agree very well with our expectations. The Hessian of the extended Rosenbrock function has $O(n)$ elements, so as expected the number of C-matrices and hence $\mathcal{D}$-updates is relatively constant for the sparse method, consistent with the bound $O(\frac{p}{n})$ for obtaining the desired C_Q-elements. In the case of the regular method, $\mathcal{D}$-updates become fewer as n grows, consistent with the bound $O(n)$ on the computation of C_Q in this case. In addition, the sparse method uses fewer function evaluations to reach the optimum, apparently since it is able to change search basis and hence adapt to the landscape of the function more often than the regular method.

On the Broyden tridiagonal function we see a similar picture, although the savings in function evaluations are not as apparent here as on the extended Rosenbrock function. The reason this seems to be that frequent basis updates is not crucial on this function. The same can be said about the results on the Broyden banded function. Note that on the two Broyden functions, when $n = 64$ and $n = 128$, no basis change takes place in the case of the regular method, which then in reality becomes Compass Search.

We also tested on the functions with noise, specifically

$$\widetilde{f}(x) = f(x) + \max(10^{-4} \cdot |f(x)|, 10^{-4}) \cdot \mu, \tag{25}$$

where μ is uniformly distributed in the interval $[-1, 1]$. This noise scheme is adopted from [18]. On these problems, the convergence criterion used was

$$\max_i \delta_i < 10^{-4}.$$

The results are listed in table 2. Since we add noise to the problems by (25) we cannot expect to find a lower function value than 10^{-4}. On the extended Rosenbrock function the picture is very much the same as with no noise. However, the regular method terminates prematurely for n equal to 32, 64, and 128. The sparse method terminates prematurely for $n = 128$. On the Broyden functions we also have the same picture as when no noise is added.

5 Concluding Remarks

We have proposed and extension to the algorithm of [7] to make it aware of sparsity, and thereby enable solution of problems with n relatively large. We have managed to reduce the number of function evaluations it takes to reach a minimum on all three test functions as n grows. The results hold promise, and much can be done to improve the results still, for

Extended Rosenbrock Function

	Sparse				Regular		
n	ρ	#feval	#Basis	f^*	#feval	#Basis	f^*
4	6	893	16	1.53e-15	1051	14	3.52e-13
8	12	1972	18	5.89e-16	2870	11	3.26e-16
16	24	3669	17	1.99e-15	8128	8	9.62e-16
32	48	7368	17	3.65e-15	20632	6	2.77e-15
64	96	14849	17	1.63e-15	65284	4	1.18e-14
128	192	29781	17	3.26e-15	190884	3	2.13e-13

Broyden Tridiagonal Function

	Sparse				Regular		
n	ρ	#feval	#Basis	f^*	#feval	#Basis	f^*
4	7	355	6	1.53e-13	365	5	4.62e-13
8	15	826	7	2.59e-13	781	3	1.20e-13
16	31	1556	6	8.25e-13	1672	2	7.97e-13
32	63	3384	7	4.09e-13	4153	1	7.52e-14
64	127	6440	7	1.70e-12	9186	0	1.42e-12
128	255	14997	8	1.41e-12	18879	0	8.61e-12

Broyden Banded Function

	Sparse				Regular		
n	ρ	#feval	#Basis	f^*	#feval	#Basis	f^*
4	10	457	6	5.08e-15	382	5	2.56e-13
8	35	824	3	1.36e-14	804	3	4.28e-13
16	91	1667	3	5.05e-14	1682	2	2.34e-13
32	203	3439	2	6.90e-13	3437	1	9.31e-13
64	427	6709	2	1.45e-12	7524	0	1.76e-13
128	875	13450	2	2.24e-12	15070	0	3.96e-13

Table 1: Numerical results, smooth functions.

Extended Rosenbrock Function

	Sparse				Regular		
n	ρ	#feval	#Basis	f^*	#feval	#Basis	f^*
4	6	808	15	3.63e-5	874	11	3.52e-4
8	12	1635	15	2.67e-3	2251	9	1.31e-4
16	24	3113	14	2.36e-2	7556	8	4.35e-3
32	48	7014	14	2.36e-2	10623	3	3.06e1
64	96	14085	16	1.38e-1	5236	0	1.22e2
128	192	29321	17	1.86e1	6629	0	2.49e2

Broyden Tridiagonal Function

	Sparse				Regular		
n	ρ	#feval	#Basis	f^*	#feval	#Basis	f^*
4	7	182	3	5.25e-5	220	3	6.71e-5
8	15	383	3	3.66e-5	400	2	9.09e-5
16	31	855	4	1.86e-4	923	1	1.98e-4
32	63	1710	4	6.69e-4	1955	0	9.15e-4
64	127	3436	4	1.03e-4	4460	0	2.03e-3
128	255	6834	4	1.70e-3	8146	0	4.81e-3

Broyden Banded Function

	Sparse				Regular		
n	ρ	#feval	#Basis	f^*	#feval	#Basis	f^*
4	10	205	3	1.73e-5	264	4	2.70e-5
8	35	460	2	5.76e-5	434	2	7.89e-5
16	91	893	1	1.13e-4	925	1	1.50e-4
32	203	1687	1	1.93e-4	1885	0	2.53e-4
64	427	3734	1	2.81e-4	3791	0	7.56e-4
128	875	6799	1	8.60e-4	7504	0	1.33e-3

Table 2: Numerical results, noisy functions.

instance incorporating ideas like the one in [16] mentioned in the introduction, and dealing with the great number of technical issues which arise when converting the algorithm of [7] to handle sparse Hessians.

References

[1] J. Borggaard, D. Pelletier, and K. Vugrin. On sensitivity analysis for problems with numerical noise. AIAA Paper 2002–5553, Presented at the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia, 2002.

[2] I. D. Coope and C. J. Price. A direct search conjugate directions algorithm for unconstrained minimization. *ANZIAM Journal*, 42(E):C478–C498, 2000.

[3] Chandler Davis. Theory of positive linear dependence. *American Journal of Mathematics*, 76:733–746, 1954.

[4] John E. Dennis Jr., Christopher J. Price, and Ian D. Coope. Direct search methods for nonlinearly constrained optimization using filters and frames. *Optimization and Engineering*, 5:123–144, 2004.

[5] C. H. Edwards. *Advanced Calculus of Several Variables*. Academic Press, 1973. ISBN 0–12–232550–8.

[6] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons Ltd., 1987. Second Edition, ISBN 0–471–91547–5.

[7] Lennart Frimannslund and Trond Steihaug. A generating set search method using curvature information. To appear, 2004.

[8] Alexader Graham. *Kronecker Products and Matrix Calculations with Applications*. Halsted Press, John Wiley and Sons, New York, 1981. ISBN 0470273003.

[9] Andreas Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia, PA, 2000. ISBN 0–89871–451–6.

[10] Andreas Griewank and Christo Mitev. Detecting jacobian sparsity patterns by bayesian probing. *Mathematical Programming*, 93(1):1–25, 2002.

[11] Andreas Griewank and Philippe L. Toint. On the unconstrained optimization of partially separable functions. In Michael J. D. Powell, editor, *Nonlinear Optimization 1981*, pages 301–312. Academic Press, New York, NY, 1982.

[12] Tamara G. Kolda, Robert Michael Lewis, and Virginia Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482, 2003.

[13] Robert Michael Lewis and Virginia Torczon. Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization*, 10(3):917–941, 2000.

[14] Robert Michael Lewis and Virginia Torczon. A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Optimization*, 12(4):1075–1089, 2002.

[15] Jorge J. Moré, Burton S. Garbow, and Kenneth E. Hillstrom. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software*, 7(1):17–41, 1981.

[16] C. P. Price and P. Toint. Exploiting problem structure in pattern search methods for unconstrained optimization. Technical Report 2004/3, Mathematics and Statistics department, Canterbury University, Christchurch, New Zealand, 2004.

[17] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, October 1960.

[18] Virginia Torczon. *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*. PhD thesis, Department of Mathematical Sciences, Rice University, Houston, Texas, 1989; available as Tech. Rep. 90-07, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77005-1892.

Duality in MIP

Generating Dual Price Functions Using Branch-and-Cut

Elena V. Pachkova

Abstract

This presentation treats duality in Mixed Integer Programming (MIP in short). A dual of a MIP problem includes a dual price function F, that plays the same role as the dual variables in Linear Programming (LP in the following).

The price function is generated while solving the primal problem. However, different to the LP dual variables, the characteristics of the dual price function depend on the algorithmic approach used to solve the MIP problem. Thus, the cutting plane approach provides nondecreasing and superadditive price functions while branch-and-bound algorithm generates piecewise linear, nondecreasing and convex price functions.

Here a hybrid algorithm based on branch-and-cut is investigated, and a price function for that algorithm is established. This price function presents a generalization of the dual price functions obtained by either the cutting plane or the branch-and-bound method.

1 Introduction

Duality in mathematical programming is used in a variety of applications. Apart from conceptual interest it provides interesting economic interpretations of the problem. Moreover, using dual information usually improves the performance of an algorithm. Thus, there exist many results on duality in linear programming (LP) (e.g. see Gass (1985)). Results on duality in integer programming (IP) also exist (Wolsey (1981)). While algorithms for LP produce unique dual programs (apart from degenerating programs), that are relatively easy to obtain, IP algorithms generate a dual function whose characteristics depend on the method used to solve the primal IP problem. Wolsey (1981) characterized this function for

the branch-and-bound and the cutting plane methods.

Duality in mixed integer programming problems (MIPs), on the other hand, has only a few results[1]. The formulation of a MIP dual also contains a dual price function as in the case of IP problems. The aim of this paper is to give a characterization of this function for the branch-and-cut method, a hybrid method, that uses the branch-and-bound and the cutting plane approaches simultaneously.

2 MIP Problems

MIP deals with models, where a linear objective function has to be maximized (or minimized) subject to a set of linear inequality or equality constraints, and where some of the variables are integer.
A classical mixed integer program can be written as:

$$\begin{array}{lll} & \max & cx + dy \\ (P_{MIP}) & s.t. & Ax + By \leq b \\ & & x \in \mathbb{Z}^n_+, y \in \mathbb{R}^m_+ \end{array} \tag{1}$$

Here, x represents the integer variables while y represents the continuous variables. $c \in \mathbb{R}^n$ and $d \in \mathbb{R}^m$ are the objective coefficients for x and y respectively. $A \in \mathbb{R}^{k \times n}$ is a $k \times n$ coefficient matrix for integer variables x and analogously $B \in \mathbb{R}^{k \times m}$ is a $k \times m$ coefficient matrix for continuous variables y. $b \in \mathbb{R}^k$ is the right hand side vector of the constraints. A review on MIP can be found in Nemhauser and Wolsey (1988).

3 Mixed Integer Duality

Consider the MIP problem (P_{MIP}) given by (1). The dual of the problem can be written as

$$\begin{array}{lll} \min & F(b) & \\ s.t. & F(Ax + By) \geq cx + dy & \forall x \in \mathbb{Z}^n_+ \quad \& \quad \forall y \in \mathbb{R}^m_+ \\ & F \in \mathfrak{F} & \end{array} \tag{2}$$

[1]Nemhauser and Wolsey (1988) have stated the dual of MIP for superadditive dual function. Nemhauser and Wolsey (1985) have investigated duality for 0-1 MIP problems.

Here F is the dual function, or the so called price function. It plays the same role as shadow prices[2] in the LP dual. Let, e.g. b be the available resources, $cx + dy$ be the profit from a production and $Ax + By$ be the production function. Then the original MIP problem (P_{MIP}) given by (1) can be interpreted as maximizing profit from production, given some constraints on available resources. One interpretation of the dual price function in the dual program (2) tells, how much extra resources are worth. In particular, if one constraint in the primal problem (1) represents a constraint on one single resource, then one extra unit of resource i is worth $F(e_i)$ units of payment, where e_i is the i'th unit vector.

The structure of an optimal price function F and its properties depend on the algorithmic approach used to solve the original MIP problem, and thus to generate F (if it is possible). A review on the pure integer programming case can be found in L. A. Wolsey (1981).

The two most widespread algorithmic approaches to solve MIP problems are branch-and-bound and cutting plane approaches. A cutting plane algorithm for MIP was first proposed by Gomory (1960). However, the procedure appeared to be slow at first. Moreover, a finite cutting plane algorithm for MIP is still not known. If the classical Gomory cuts are used, Salkin (1989) mentions an example of a MIP problem by White (1961), that cannot be solved using the cutting plane method. Therefore the research was more concentrated on the branch-and-bound method proposed by Little (1963).

However, the cutting planes algorithms have been reconsidered in the early 90's with some impressive results. Thus, a cutting plane based lift-and-project algorithm was proposed (see Balas et al. (1993) and Lovasz and Schrijver (1991)). Moreover, one of the most widespread algorithm, branch-and-cut[3] is a mixture of both approaches where a cutting plane approach is added to the branch-and-bound framework.

Two sets of functions will be useful when describing MIP problems. Let $\mathfrak{F}$ be the set of nondecreasing functions $F : \mathbb{R}^k \to \mathbb{R}$. Thus

$$\mathfrak{F} = \{(F : \mathbb{R}^k \to \mathbb{R}) : F(a) \leq F(b) \forall a, b \in \mathbb{R}^k, a \leqq b\}.$$

Finally let $\mathfrak{H}$ be the set of nondecreasing and superadditive functions satisfying the following conditions:

1. $(F : \mathbb{R}^k \to \mathbb{R}) \in \mathfrak{H}$ is superadditive, i.e. $F(q_1) + F(q_2) \leq F(q_1 + q_2), \forall q_1, q_2 \in \mathbb{R}^k$,

[2]Dual variables

[3]Padberg and Rinaldi (1987) for pure integer programming and Crowder et al. (1983) for 0-1 MIP.

2. $F(\mathbf{0}) = 0$,

3. $F \in \mathfrak{H}$ is nondecreasing, i.e. $F \in \mathfrak{F}$, and

4. $\overline{F}(q) = \lim_{\epsilon \searrow 0} \frac{F(\epsilon q)}{\epsilon}$ exists and is finite for all q.

In the cutting plane approach we will deal with functions in $\mathfrak{H}$, while dual price functions for branch-and-bound approach will be nondecreasing, polyhedral and convex.

3.1 Cutting Plane Framework

Algorithms based on the cutting plane setting are less common. This may be because no cutting plane based finite algorithm is known for the general MIP problem. See Marchand et al. (1999) for a review on cutting plane based algorithms for MIP problems. The original Gomory's cutting plane algorithm is sure to terminate only if the optimal objective function is integer valued. Other MIP algorithms restrict the variables to the 0-1 case.

Again consider the MIP problem (P_{MIP}) given by (1). The Gomory's strong cutting plane algorithm for MIP problems solves a family of problems (P^r):

$$\begin{aligned} \max \quad & z^r = cx + dy \\ s.t \quad & Ax + By \leq b \\ & Cx + \overline{C}y \leq C_b \\ & x, y \geq \mathbf{0} \end{aligned}$$

Here an element in the last set of constraints has the form $\sum_{j=1}^{n} G_r(A_{.j})x_j + \sum_{j=1}^{m} \overline{G}_r(B_{.j})y_j \leq G_r(b)$ where the function $G_r(q) : \mathbb{R}^k \to \mathbb{R}$ represents a Gomory cut. r is the index representing the number of the cut in focus.

The form of the function $G_r(q)$ can be obtained from results in Nemhauser and Wolsey (1988). Let $\lfloor a \rfloor$ be the integral part, and f_a be the fractional part of $a \in \mathbb{R}$. That is, $a = \lfloor a \rfloor + f_a$ and $0 \leq f_a \leq 1$. For an α, $0 \leq \alpha < 1$, define $F_\alpha(a) : \mathbb{R} \to \mathbb{R}$ by

$$F_\alpha(a) = \lfloor a \rfloor + \max(0, \tfrac{f_a - \alpha}{1 - \alpha}).$$

Let v be the row element of the inverse basis matrix corresponding to the source row in the constructed simplex tableau. For simplicity consider the first cut. Then $v = \{v_1, ..., v_k\}$, since the dimension of the basis is k. Let $V = \{1, ..., k\}$, $V^+ = \{i \in V | v_i \geq 0\}$ and

$V^- = \{i \in V | v_i < 0\}$. Moreover, let α be the fractional part of the nonintegral value of the basic variable in the source row.

Holm and Tind (1988)[4] show that $G(q)$ defined by

$$G(q) = F_\alpha(vq) - \frac{1}{1-\alpha} \sum_{i \in V^-} v_i q_i$$

is superadditive and nondecreasing and that

$$\overline{G}(q) = \frac{1}{1-\alpha} \min(-\sum_{i \in V^-} v_i q_i, \sum_{i \in V^+} v_i q_i)$$

is concave and piecewise linear. Additionally $G(q)$ generates cuts in the Gomory strong MIP cutting plane algorithm.

The algorithm terminates if some problem (P^r) is found to be infeasible, or if a mixed integer solution is found. However, this Gomory cutting plane algorithm is finite for integral optimum objectives only. For a general MIP we are not sure to obtain a solution after a finite number of cuts.

3.1.1 MIP duality in Cutting Plane Framework

Gomory's strong mixed integer cutting plane algorithm generates nondecreasing superadditive optimal dual price functions. Suppose that p Gomory cuts are needed to find the optimal solution for the primal MIP problem. With the Gomory cuts given by the function $G(q)$ defined above, the optimal price function $F(q) : \mathbb{R}^k \to \mathbb{R}$ and its directional derivative are given by

$$F(q) = \sum_{i=1}^{k} u_i q_i + \sum_{i=k+1}^{k+p} u_i G_i(q) \tag{3}$$

and

$$\overline{F}(q) = \sum_{i=1}^{k} u_i q_i + \sum_{i=k+1}^{k+p} u_i \overline{G}_i(q)$$

respectively. Here $u_1, ..., u_k, u_{k+1}, ..., u_{k+p} \geq 0$ represent the dual variables obtained at termination. The first k variables correspond to the original MIP constraints, while the last

[4]Based on Nemhauser and Wolsey (1988)

p variables correspond to the additional Gomory cuts.

The superadditive dual of a MIP is then (see also Nemhauser and Wolsey (1988))

$$\begin{aligned} \min_{F \in \mathfrak{H}} & F(b) \\ s.t.\ F(A_{.j}) &\geq c_j \quad j = 1, ..., n \\ \overline{F}(B_{.j}) &\geq d_j \quad j = 1, ..., m \end{aligned}$$

Here $F(q)$ is nondecreasing and superadditive and $\overline{F}(q)$ is concave and piecewise linear.

3.2 Branch-and-Bound Framework

The LP based branch-and-bound approach produced some effective algorithms like branch-and-price and branch-and-cut. A review on algorithms based on LP branch-and-bound approach can be found in Johnson et al. (2000).

Consider the mixed integer problem (P_{MIP}) given by (1). The classical branch-and-bound algorithm solves a family of subproblems (P_t), $t = 1, ..., r$:

$$\begin{aligned} \max \quad & cx + dy \\ s.t. \quad & Ax + By \leq b \\ & x \in X_t, y \in \mathbb{R}^m_+ \end{aligned} \tag{4}$$

where $\mathbb{Z}^n_+ \subseteq \bigcup_{t=1}^r X_t$. Assume in the following that $X_t = \{x \in \mathbb{R}^n : g^t_j \leq x_j \leq h^t_j, j = 1, ..., n, x \geq \mathbf{0}\}$ as it is done in Klamroth et al. (2002), where g^t_j and h^t_j are lower and upper integer bounds respectively. This assumption is satisfied by LP based branch-and-bound approaches and many other branch-and-bound algorithms. A branch-and-bound algorithm terminates if one of the following is true:

- All the generated subproblems (P_t), $t = 1, ..., r$, are shown to be infeasible or,
- The optimal solution to some subproblem P_{t^*}, (x^{t^*}, y^{t^*}), is found, such that x^{t^*} is integer valued, and for $z_{t^*} = cx^{t^*} + dy^{t^*}$ we have that $z_{t^*} \geq z_t$ for all $t \neq t^*$. Here z_t represents the objective value of the subproblem (P_t).

3.2.1 MIP Duality in Branch-and-Bound Framework

Using branch-and-bound algorithms the generated optimal price function is not necessarily superadditive. Although branch-and-bound algorithms have been widespread in solving MIP problems, there are no results concerning generation of optimal price functions based on branch-and-bound known to the author. A treatment for the pure integer programming problem can be found in Wolsey (1981).

Consider the original MIP problem (P_{MIP}) given by (1) and the subproblems (P_t) given by (4). The following lemma shows how to construct a dual feasible function for (P_{MIP}) given dual feasible functions for its subproblems.

Lemma 3.1: *If* $F_t \in \mathfrak{F}, \quad t = 1, ..., r,$ *are dual feasible functions for the subproblems* $(P_t), \quad t = 1, ..., r$ *in the sense that*

$$F_t(Ax + By) \geq cx + dy \quad \forall x \in X_t, y \in \mathbb{R}^m_+$$

then

$$F(q) := \max_{t=1,...,r} F_t(q)$$

is a dual feasible function for the original MIP problem (P_{MIP}) *in (1).*

Proof

Let $x \in \mathbb{Z}^n_+$, and $y \in \mathbb{R}^m_+$. Then because $\mathbb{Z}^n_+ \subseteq \bigcup_{t=1}^r X_t$, $x \in X_t$ for some $t = 1, ..., r$. Hence, since F_t is feasible for (P_t), $F_t(Ax + By) \geq cx + dy$. But due to the definition of F, $F(Ax + By) \geq F_t(Ax + By) \geq cx + dy$. Moreover, F is nondecreasing since F_t is nondecreasing for $t = 1, ..., r$. This implies that $F \in \mathfrak{F}$. Thus, all in all F is a dual feasible function for the original MIP problem (P_{MIP}).

□

Next we show that a dual optimal function F for the original MIP problem in fact exists, provided the problem has a finite optimal solution. This result together with a way to construct F is established in the theorem below.

Theorem 3.1 *If the original MIP program* (P_{MIP}) *in (1) has a final optimal solution, and an LP based branch-and-bound algorithm terminates in a finite number of subproblems* (P_t), $t = 1, ..., r,$ *then there exists a dual optimal price function* $F \in \mathfrak{F}$ *where*

$$F(q) := \max_{t=1,...,r} (\pi^t q + \alpha^t), \quad \alpha^t \in \mathbb{R}, \pi^t \in \mathbb{R}^k, \pi^t \geq \mathbf{0}. \tag{5}$$

Proof

Let z^* be the optimum objective value of (P_{MIP}) and consider some arbitrarily chosen terminating subproblem (P_t), $t \in 1, ..., r$. (P_t) is either infeasible or has an optimal solution where integer variables have integer values.

a) If the linear program (P_t) has an appropriate optimal solution with corresponding optimal objective value z_t, then its dual LP is feasible. Let $(\pi^t, \underline{\pi}^t, \overline{\pi}^t) \geq \mathbf{0}$ be the optimal solution of the dual. Here, the variable π^t corresponds to the initial constraints in (P_{MIP}), while the variables $\underline{\pi}^t$ and $\overline{\pi}^t$ represent the extra integer $\geq$ and $\leq$ constraints respectively, that are generated by the branch-and-bound algorithm. Since $(\pi^t, \underline{\pi}^t, \overline{\pi}^t)$ is feasible for the dual LP

$$\pi^t A_{.j} - \sum_{j=1}^{n} \underline{\pi}_j^t + \sum_{j=1}^{n} \overline{\pi}_j^t \geq c_j \quad j = 1, ..., n$$

and

$$\pi_t B_{.i} \geq d_i \quad i = 1, ..., m.$$

Define a nondecreasing function F_t as

$F_t(q) := \pi^t q + \alpha^t$, where $\alpha^t = -\underline{\pi}^t g^t + \overline{\pi}^t h^t$.

F_t satisfies

$F_t(Ax + By) = \pi^t(Ax + By) + \alpha^t = \pi^t(Ax + By) - \underline{\pi}^t g^t + \overline{\pi}^t h^t \geq$
$\pi^t(Ax + By) - \underline{\pi}^t x + \overline{\pi}^t x = \pi^t Ax + \pi^t By - \underline{\pi}^t x + \overline{\pi}^t x \geq cx + dy \; \forall x \in X_t, y \in \mathbb{R}^m$.

Thus, F_t represents a dual feasible function for (P_t) in the sense of lemma 3.1. Moreover, by linear programming duality, $F_t(b) = \pi^t b - \underline{\pi}^t g^t + \overline{\pi}^t h^t = z_t$ for terminating (P_t). Here $z_t \leq z^*$.

b) If (P_t) on the other hand is infeasible, there exits a dual ray $(\omega^t, \underline{\omega}^t, \overline{\omega}^t) \geq \mathbf{0}$, that satisfies $\omega^t A_{.j} - \sum_{j=1}^{n} \underline{\omega}_j^t + \sum_{j=1}^{n} \overline{\omega}_j^t \geq c_j, \quad j = 1, ..., n, \; \omega_t B_{.i} \geq d_i, \quad i = 1, ..., m$ and $\omega^t b - \underline{\omega}^t g^t + \overline{\omega}^t h^t < 0$. The definitions of ω are analogous to the definitions of π above. Consider some dual feasible solution $(\pi^p, \underline{\pi}^p, \overline{\pi}^p) \geq \mathbf{0}$ of the dual of (P_t). This may be available from the parent node in the branch-and-bound tree. Combining it with the dual ray we obtain a vector $(\pi^t, \underline{\pi}^t, \overline{\pi}^t) := (\pi^p, \underline{\pi}^p, \overline{\pi}^p) + \mu(\omega^t, \underline{\omega}^t, \overline{\omega}^t)$, where $\mu \in \mathbb{R}_+$.

Define $F_t \in \mathfrak{F}$ for (P_t) by $F_t(q) = \pi^t q + \alpha^t$, $\alpha^t := -\underline{\pi}^t g^t + \overline{\pi}^t h^t$. Then we have that:

$F_t(Ax + By) = \pi^t(Ax + By) - \underline{\pi}^t g^t + \overline{\pi}^t h^t =$
$(\pi^p + \mu\omega^t)(Ax + By) - (\underline{\pi}^p + \mu\underline{\omega}^t)g^t + (\overline{\pi}^p + \mu\overline{\omega}^t)h^t =$
$\pi^p(Ax + By) + \mu\omega^t(Ax + By) - \underline{\pi}^p g^t - \mu\underline{\omega}^t g^t + \overline{\pi}^p h^t + \mu\overline{\omega}^t h^t \geq cx + dy, \forall x \in X_t, y \in \mathbb{R}^m_+$

Thus again we are dealing with a dual feasible function F_t for (P_t). Moreover, we see that

$$\lim_{\mu\to\infty} F_t(b) = \lim_{\mu\to\infty}(\pi^t b + \alpha^t) = \lim_{\mu\to\infty}((\pi^p + \mu\omega^t)b - (\underline{\pi}^p + \mu\underline{\omega}^t)g^t + (\overline{\pi}^p + \mu\overline{\omega}^t)h^t) =$$
$$\lim_{\mu\to\infty}(\pi^p b - \underline{\pi}^p g^t + \overline{\pi}^p h^t + \mu(\omega^t b - \underline{\omega}^t g^t + \overline{\omega}^t h^t)) = -\infty.$$

Thus we always can choose μ so $F_t(b) < z^*$.

Summarizing, $F_t(q) = \pi^t q + \alpha^t$ is a dual feasible function for all terminating (P_t), $t = 1, ..., r$. Thus, using lemma 3.1, the price function F given by (5), is dual feasible for (P_{MIP}). We assumed that (P_{MIP}) has an finite optimal mixed integer solution, let it be (x^*, y^*). But then there exists a $t^* \in \{1, ..., r\}$ such that (x^*, y^*) is the optimum solution for (P_{t^*}) and hence $z^* = cx^* + dy^* = F_{t^*}(b)$. Since $F_t(b) \leq z^*$ for all $t = 1, ..., r$, $F(b) = \max_{t=1,...,r} F_t(b) = F_{t^*}(b) = z^*$ and thus is dual optimal for (P_{MIP}). All in all, the constructed optimal dual price function F exists and is dual optimal for (P_{MIP}).

□

The theorem shows that a standard LP based branch-and-bound algorithm generates a price function that is piecewise linear, nondecreasing and convex, as it was the case with pure IP problems (see Wolsey (1981)). We also see that F in general is not superadditive.

There are several versions of the branch-and-bound algorithms, depending on which variable to branch on, if several integer variables have non-integer values in an optimal solution of a LP relaxation. Each version produces one optimal dual price function. Thus, the generated price function is only one possible solution out of many and depends on the version of the algorithm.

For a special kind of MIP problems, however, an interpretation involving a superadditive price function can be obtained using branch-and-bound algorithms. An analogous result for the pure integer programming case can be found in Wolsey (1981). Consider the

following bounded MIP problem $(\overline{P})$:

$$\begin{array}{ll} \max & cx + dy \\ s.t. & Ax + By \leq b \\ & -x \leq -g \\ & x \leq h \\ & x \in \mathbb{Z}^n_+, y \in \mathbb{R}^m_+ \end{array}$$

Let again $X_t = \{x \in \mathbb{R}^n : g^t_j \leq x_j \leq h^t_j, j = 1, ..., n, x \geq \mathbf{0}\}$ and $\{x : \mathbf{0} \leq g \leq x \leq h, x \geq \mathbf{0}$ and integer$\} \subseteq \bigcup_{t=1}^r X_t$. The dual of $(\overline{P})$ is

$$\begin{array}{lrcl} \min & F(b, -g, h) & & \\ s.t. & F(A_{.j}, -e_j, e_j) & \geq & c_j \\ & \overline{F}(B_{.j}, 0, 0) & \geq & d_j \\ & F \in \mathfrak{H} & & \end{array}$$

where e_j is the j'th unit vector.

Theorem 3.2 *If the bounded MIP program $(\overline{P})$ has a final optimal solution, and solving $(\overline{P})$ with an LP based branch-and-bound algorithm results in a finite number of terminating subproblems (P_t), $t = 1, ..., r$, then there exists a dual feasible price function $F \in \mathfrak{H}$ of the form*

$$F(q) := \min_{t=1,...,r} u^t q, \quad u^t \in \mathbb{R}^{k+2n}, u^t \geq \mathbf{0}.$$

Proof

Set $u^t = (\pi^t, \underline{\pi}^t, \overline{\pi}^t)$, as in the proof of theorem 3.1. Thus, u^t is the dual variables of some subproblem $(\overline{P}_t)$ in case a) and a combination of a feasible solution and a dual ray in case b). Since $\pi^t A_{.j} - \underline{\pi}^t_j + \overline{\pi}^t_j \geq c_j$ for all $t = 1, ..., r$, $F(A_{.j}, -e_j, e_j) = \min_{t=1,...,r}(\pi^t A_{.j} - \underline{\pi}^t_j + \overline{\pi}^t_j) \geq c_j$. Moreover, since $\pi^t B_{.j} \geq d_j$ for all $t = 1, ..., r$, $\overline{F}(B_{.j}, 0, 0) = \min_{t=1,...,r}(\pi^t B_j) \geq d_j$.

F is clearly superadditive and nondecreasing and $F(\mathbf{0}) = 0$. Finally finite $\overline{F}(q)$ exists for all q. Thus, $F \in \mathfrak{H}$. All in all, F is dual feasible for $(\overline{P})$.

□

The generated price function is a weak dual function and serves as an upper bound for the value function of the primal problem $(\overline{P})$.

3.3 MIP Duality in Branch-and-Cut Framework

The branch-and-cut algorithm is a hybrid algorithm, that combines the branch-and-bound and the cutting planes approaches. Thus, at each node we try to find a violated cut first. If it is not available within a reasonable amount of time we branch. A description of the algorithm can among others be found in Cordier et al. (1999). This algorithm turned out to be quite effective for solving MIP problems.

The following theorem states a result about the dual optimal function of the MIP problem, if an branch-and-cut algorithm is applied. Such a dual function exists provided that the primal problem has a finite optimal solution, and the number of terminating subproblems is finite. Moreover, the theorem shows a way to find an optimal dual price function.

Theorem 3.3 *If the original MIP program* (P_{MIP}) *in (1) has a final optimal solution, and solving* (P_{MIP}) *with a branch-and-cut algorithm results in a finite number of terminating subproblems* $(\tilde{P}_t)$*,* $t = 1, ..., r$*, then there exists a dual optimal price function* $F \in \mathfrak{F}$ *where*

$$F(q) := \max_{t=1,...,r} (\pi^t q + \alpha^t + \sum_{s=1}^{\delta(t)} \tilde{\pi}_s^t G_s^t(q)), \quad \alpha^t \in \mathbb{R}, \pi^t \in \mathbb{R}_+^k, \tilde{\pi}^t \in \mathbb{R}_+^{\delta(t)}, G_s^t \in \mathfrak{H}.$$

Here $\delta(t) \geq 0$ *is the number of Gomory cuts* G_s^t *in subproblem* $(\tilde{P}_t)$*.*

Proof

As in the proof of theorem 3.1 let z^* be the optimum objective value of (P_{MIP}). Consider some arbitrarily chosen terminating subproblem $(\tilde{P}_t)$:

$$\begin{array}{ll} \max & cx + dy \\ s.t. & Ax + By \leq b \\ & Cx + \overline{C}y \leq C_b \\ & x \in X_t, y \in \mathbb{R}_+^m \end{array}$$

where an element in the last constraints has the form $\sum_{j=1}^n G_s^t(A_{.j})x_j + \sum_{j=1}^m \overline{G}_s^t(B_{.j})y_j \leq G_s^t(b)$, and $G_s^t(q)$ represents the s'th Gomory cut in problem $(\tilde{P}_t)$. If some cuts are present in a parent node subproblem, then these cuts will also be present in its child node subproblem, if such a child node exists.

case a) Suppose that the LP relaxation of $(\tilde{P}_t)$ has an optimal mixed integer solution with objective value z_t. Let $(\pi^t, \overline{\pi}^t, \underline{\pi}^t, \tilde{\pi}^t) \geq \mathbf{0}$ be the optimal dual solution. $\pi^t, \overline{\pi}^t, \underline{\pi}^t$ are as defined in proof for theorem 3.1, and $\tilde{\pi}^t$ corresponds to the Gomory cuts constraints. Since we have $\delta(t)$ cuts in problem $(\tilde{P}_t)$, $\tilde{\pi}^t$ has dimension $\delta(t)$. We see that

$$\pi^t A_{.j} - \sum_{j=1}^{n} \underline{\pi}_j^t + \sum_{j=1}^{n} \overline{\pi}_j^t + \sum_{s=1}^{\delta(t)} G_s^t(A_{.j})\tilde{\pi}_s^t \geq c_j \quad j = 1, ..., n$$

and

$$\pi^t B_{.i} + \sum_{s=1}^{\delta(t)} \overline{G}_s^t(B_{.i})\tilde{\pi}_s^t \geq d_i \quad i = 1, ..., m.$$

Define the nondecreasing function F_t by

$F_t(q) := \pi^t q + \alpha^t + \sum_{s=1}^{\delta(t)} G_s^t(q)\tilde{\pi}_s^t$ with $\alpha^t = -\underline{\pi}^t g^t + \overline{\pi}^t h^t, t = 1, ..., r.$

Since G_s^t is a Gomory cut it is superadditive. But then $\sum_{s=1}^{\delta(t)} G_s^t(Ax + By)\tilde{\pi}_s^t \geq$ $\sum_{s=1}^{\delta(t)} G_s^t(Ax)\tilde{\pi}_s^t + \sum_{s=1}^{\delta(t)} G_s^t(By)\tilde{\pi}_s^t$. Moreover,this also implies that $\sum_{s=1}^{\delta(t)} G_s^t(Ax)\tilde{\pi}_s^t \geq$ $\sum_{s=1}^{\delta(t)} G_s^t(A)x\tilde{\pi}_s^t$, and analogously $\sum_{s=1}^{\delta(t)} G_s^t(By)\tilde{\pi}_s^t \geq \sum_{s=1}^{\delta(t)} G_s^t(B)y\tilde{\pi}_s^t$. Due to the definition of G_s^t given in section 3.1 $G_s^t(B) \geq \overline{G}_s^t(B)$, $s = 1, ..., \delta(t)$. Finally, since G_s^t is a Gomory cut $G_s^t(0) = 0$.

All this implies that, for all $t = 1, ..., r$, F_t satisfies

$F_t(Ax + By) = \pi^t(Ax + By) + \overline{\pi}^t h^t - \underline{\pi}^t g^t + \sum_{s=1}^{\delta(t)} G_s^t(Ax + By)\tilde{\pi}_s^t \geq$
$\pi^t Ax + \pi^t By + \overline{\pi}^t x - \underline{\pi}^t x + \sum_{s=1}^{\delta(t)} G_s^t(Ax)\tilde{\pi}_s^t + \sum_{s=1}^{\delta(t)} G_s^t(By)\tilde{\pi}_s^t \geq$
$\pi^t Ax + \pi^t By + \overline{\pi}^t x - \underline{\pi}^t x + \sum_{s=1}^{\delta(t)} G_s^t(A)x\tilde{\pi}_s^t + \sum_{s=1}^{\delta(t)} G_s^t(B)y\tilde{\pi}_s^t \geq$
$\pi^t Ax + \pi^t By + \overline{\pi}^t x - \underline{\pi}^t x + \sum_{s=1}^{\delta(t)} G_s^t(A)x\tilde{\pi}_s^t + \sum_{s=1}^{\delta(t)} \overline{G}_s^t(B)y\tilde{\pi}_s^t \geq$
$cx + dy \quad \forall x \in X_t, y \in \mathbb{R}^m.$

Thus, the function F_t represents a dual feasible function for $(\tilde{P}_t)$. Moreover, by linear programming duality, $F_t(b) = \pi^t b - \underline{\pi}^t g^t + \overline{\pi}^t h^t + \sum_{s=1}^{\delta(t)} G_s^t(b)\tilde{\pi}_s^t = z_t$ for terminating $(\tilde{P}_t)$ and $z_t \leq z^*$.

case b) If $(\tilde{P}_t)$ is infeasible then there exists a dual ray $(\omega^t, \underline{\omega}^t, \overline{\omega}^t, \tilde{\omega}^t) \geq \mathbf{0}$, such that $\omega^t A_{.j} - \sum_{j=1}^{n} \underline{\omega}_j^t + \sum_{j=1}^{n} \overline{\omega}_j^t + \sum_{s=1}^{\delta(t)} G_s^t(A_{.j})\tilde{\omega}_s^t \geq c_j, j = 1, ..., n$, $\pi^t B_{.i} + \sum_{s=1}^{\delta(t)} \overline{G}_s^t(B_{.i})\tilde{\omega}_s^t \geq$ $d_i, i = 1, ..., m$, and $\omega^t b - \underline{\omega}^t g^t + \overline{\omega}^t h^t + \sum_{s=1}^{\delta(t)} G_s^t(b)\tilde{\omega}_s^t < 0$. Analogous to the proof for

theorem 3.1 define

$(\pi^t, \underline{\pi}^t, \overline{\pi}^t, \tilde{\pi}^t) := (\pi^p, \underline{\pi}^p, \overline{\pi}^p, \tilde{\pi}^p) + \mu(\omega^t, \underline{\omega}^t, \overline{\omega}^t, \tilde{\omega}^t)$, where $\mu \in \mathbb{R}_+$.

Here, $(\pi^p, \underline{\pi}^p, \overline{\pi}^p, \tilde{\pi}^p)$ represents a dual feasible solution for $(\tilde{P}_t)$. Then let F_t be defined by $F_t(q) := \pi^t q + \alpha^t + \sum_{s=1}^{\delta(t)} G_s^t(q)\tilde{\pi}_s^t$, $\alpha^t := -\underline{\pi}^t g^t + \overline{\pi}^t h^t$. Analogous to case a) F_t satisfies $F_t(Ax + By) \geq cx + dy, \forall x \in X_t, y \in \mathbb{R}_+^m$. Thus F_t represents a dual feasible function for $(\tilde{P}_t)$.

Moreover, since

$$\lim_{\mu\to\infty} F_t(b) = \lim_{\mu\to\infty} (\pi^t b + \alpha^t + \sum_{s=1}^{\delta(t)} G_s^t(q)\tilde{\pi}_s^t) =$$

$$\lim_{\mu\to\infty} (\pi^p b - \underline{\pi}^p g^t + \overline{\pi}^p h^t + \sum_{s=1}^{\delta(t)} G_s^t(b)\tilde{\pi}_s^p + \mu(\omega^t b - \underline{\omega}^t g^t + \overline{\omega}^t h^t + \sum_{s=1}^{\delta(t)} G_s^t(b)\tilde{\omega}_s^t)) = -\infty.$$

we always can choose μ so $F_t(b) < z^*$.

Summarizing, $F_t(q) = \pi^t q + \alpha^t + \sum_{s=1}^{\delta(t)} G_s^t(q)\tilde{\pi}_s^t$ is a dual feasible function for all terminating $(\tilde{P}_t), t = 1, ..., r$. Thus, $F(q) := \max_{t=1,...,r} F_t(q)$ is a dual feasible function for (P_{MIP}), if branch-and-cut algorithm is used due to lemma 3.1. Since we assumed that there is a finite mixed integer solution (x^*, y^*) there exists $t^* \in \{1, ..., r\}$ such that (x^*, y^*) is the optimal solution for $(\tilde{P}_{t^*})$. Hence $z^* = cx^* + dy^* = F_{t^*}(b)$. Since $F_t(b) \leq z^*$ for all $t = 1, ..., r$, $F(b) = F_{t^*} = z^*$ and this $F(q)$ is optimal for (P_{MIP}).

□

The constructed function is not only the dual optimal function for the branch-and-cut algorithm. It also represents a general form of such a dual optimal function if either a cutting plane or branch-and-bound based algorithm is used. In case of a cutting plane algorithm, we only deal with one node, the root node. Moreover, the variables α^t disappear. Thus we end up with the same formulation as (3). For a pure branch-and-bound algorithm, we do not have any constraints representing Gomory cuts, $\delta(t) = 0$, for all $t = 1, ..., r$. In this case we are back to the same formulation of the dual price function (5) in theorem 3.1.

4 Summary

The presented paper gave a short presentation of the MIP problem and three of its solution methods. Additionally, some duality results were shown. In particular, the formulation of a dual of a MIP problem contains a dual price function F. The characteristics of this function, however, depend on the algorithm used to generate it. Applying the cutting plane algorithm we obtain a nondecreasing and superadditive price function. Using a branch-and-bound algorithm, on the other hand, provides a piecewise linear, nondecreasing and convex price function, which in general is not superadditive. However, section 3.2.1 presented a superadditive weak dual price function for the bounded MIP problem, if branch-and-bound approach is applied.

Section 3.3 presents a general dual function for the branch-and-bound and the cutting plane approach. This dual function is additionally the price function for the branch-and-cut algorithm. The branch-and-cut algorithm is now very popular when solving MIP problems. One important brick in the algorithm is the generation of cuts. In this chapter we used the classical Gomory cut. However, there exist other cuts, e.g. the lift-and-project cut (see Balas et al. (1993), Balas et al. (1996)) or the mixed integer rounding cut (see Marchand and Wolsey (2001)). One idea for further research would be to show similar duality results for these cuts.

Apart from the conceptual interest, the result can be useful in economic interpretations of MIP models as well as sensitivity analysis.

References

E. Balas, S. Ceria, G. Cornuejols, "A lift-and-project cutting plan algorithm for mixed 0-1 programs", Mathematical Programming 58, pp. 295-324, 1993.
E. Balas, S. Ceria, G. Cornuejols, "Mixed 0-1 Programming by Lift-and-Project in Branch-and-Cut Framework", Management Science 42, 1996.
C. Cordier, H. Marchand, R. Laundy, L.A. Wolsey, "A branch-and-cut code for mixed integer programming", Mathematical Programming 86, pp. 335-353, 1999.
H. Crowder, E.L. Johnson, M.W. Padberg, "Solving Large Scale Zero-One Linear Programming Problems", Operation Research 31, pp. 803-834, 1983.
S.I. Gass, "Linear Programming", 5th ed.,McGraw-Hill, (1985).
R.E. Gomory, "An algorithm for the mixed integer problem", RM-2597, The Rand Corporation, 1960.
S. Holm, J. Tind, " A Unified Approach for Price Directive Decomposition Procedures in

Integer Programming", Discrete Applied Mathematics 20, pp. 205-219, 1988.
E. L. Johnson, Georg L. Nemhauser, Martin W. P. Savelsbergh, "Progress in linear Programming Based Branch-and-Bound Algorithms: An Exposition", INFORMS Journal on Computing 12, 2000.
K.Klamroth, J. Tind, S. Zust, "Integer Programming Duality in Multiple Objective Programming", University of Copenhagen, Department of Applied Mathematics and Statistics, 2002.
J.D.C. Little, K.G. Murty, D.W. Sweeney and C. Karel, "An Algorithm for the Travelling Salesman Problem", Operation Research 11, pp. 972-989, 1963.
L. Lovasz and A. Schrijver, "Cones of matrices and set functions and 0-1 optimization", SIAM Journal on Optimization 1(2), pp. 166-190, 1991
H. Marchand, A. Martin, R. Weismantel, L.A. Wolsey, "Cutting Planes in Integer and Mixed Integer Programming", CORE Discussion Paper 9953, 1999.
H. Marchand, L.A. Wolsey, "Aggregation and Mixed Integer Rounding to Solve MIPs", Operation Research, Vol. 49, No. 3, pp. 363-371, 2001.
G. L. Nemhauser, L. A. Wolsey, "A Recursive Procedure to Generate all Cuts for 0-1 Mixed Integer Programs", Mathematical Programming 46, pp. 379-390, 1990.
G. L. Nemhauser, L. A. Wolsey, "Integer and Combinatorial Optimization", John Wiley & Sons, Inc, 1999.
M. Padberg, G. Rinaldi, "Optimization of a 537-city TSP by branch and cut", Operations Research Letters 6, pp. 1-8, 1987.
H.M. Salkin, K. Mathur, "Foundations of Integer Programming", North-Holland, 1989.
W. White, "On Gomory's Mixed Integer Algorithm", Senior Thesis, Princeton University, May 1961.
L.A. Wolsey, "Integer Programming Duality: Price Functions and Sensitivity Analysis", Mathematical Programming, 20, pp. 173-195, 1981.

Abstracts

On the Facial Structure of the Replacement Polytope

Andréasson, Niclas *(Chalmers University of Technology, Sweden)*

Consider a system consisting of a finite number of parts, each with a specific lifetime. At the very latest when a part reaches its lifetime it must be replaced. Associated with a replacement is the cost of the part and a fixed cost independent of how many parts that are replaced. The replacement problem refers to finding a replacement schedule that minimizes the total cost for having a working system a finite time period. An integer linear program is presented for the replacement problem. The facial structure of the convex hull of the set of feasible solutions (the replacement polytope) is then investigated.

Optimization and Evolutionary Search: Related Issues

Bhattacharya, Maumita *(Charles Sturt University, Australia)*

Evolutionary algorithms (EA) have been long accepted as efficient global optimizers. Given a search space *S* and an objective function *g* defined on it, the problem is to find the global maximum (or minimum) of *g* in *S*. To apply EA's heuristic search, the coding function or representation ρ is created, that partially maps *S* to the finite chromosome space *C*. The genetic operators are used to create new solutions such that $C^n \rightarrow C^m$.
However, as the evolutionary search progresses, it is important to avoid reaching a state where the genetic operators can no longer produce superior offspring, prematurely. This is likely to occur when the search space reaches a homogeneous or near-homogeneous configuration converging to a local optimal solution. Maintaining a certain degree of population diversity is widely believed to help curb this problem. This paper discusses the problem of premature convergence related to EA based optimization. A novel technique is presented, that uses informed genetic operations to reach promising, but un/under-explored areas of the search space, while discouraging local convergence, to curb premature convergence. Elitism is used at a different level aiming at convergence. The proposed technique's improved performance in terms solution precision and convergence characteristics is observed on a number of benchmark test functions with a genetic algorithm (GA) implementation.

Determining the Non-Existence of a Compatible OSPF Metric

Broström, Peter *(Linköping Institute of Technology, Sweden)*
Holmberg, Kaj *(Linköping Institute of Technology, Sweden)*

Many communication networks use the intra-domain protocol OSPF (Open Shortest Path First) for deciding the routing of traffic. Routers in such networks send traffic to destinations on shortest paths. The network operator control the traffic by assigning weights to each link. This set of weights is called the "metric" and is used in the shortest path computations.
It is easy to decide how traffic is routed when a network and a metric is given (this is in fact exactly what routers do). A more difficult question is whether or not there exists a metric giving a set of desired traffic patterns i.e. a metric making the desired paths shortest. Such a metric is in this work called compatible. The existence of a compatible metric is a matter of similarities between different traffic patterns, and this is further investigated in this work.
To this point, there is one known necessary condition for the existence of a compatible metric, called the "sub-optimality" condition. We present more general necessary conditions for the existence of a compatible metric for a set of desired shortest path graphs. In addition, we also present a polynomial method that use pairs of traffic patterns for explaining why some desired sets are not compatible with any metric. This method is successful in indicating where the conflict lie in most instances, but can sometimes fail when the type of conflict is more complicated. More complicating conflicts are treated in the presentation "Stronger necessary conditions for the existence of an compatible OSPF metric".

On the Use of Second Derivatives in Optimization of Radiation Therapy

Carlsson, Fredrik *(Royal Institute of Technology, Sweden)*
Forsgren, Anders *(Royal Institute of Technology, Sweden)*

The goal of external-beam radiation therapy of cancer is to obtain an acceptable balance between tumor control and complications to the normal tissue surrounding the tumor. During the last decade, the field has experienced a rapid progress. New technology has improved the accuracy of the beam delivery significantly. Together with the development of faster computers, this has led the way for so called 'intensity modulated radiation therapy' (IMRT).
In IMRT, the clinician specifies certain characteristics of the desired dose distribution by introducing objective functions for the tumor and for the critical organs close to the tumor. A discretization of the incident beams and of the treatment volume of the patient is performed and an optimization problem is formulated. In general, the IMRT problem is large-scale and has a non-convex nature, often with linear and non-linear constraints.
In this study we investigate how the Hessian affects the optimization performance for a quasi-Newton algorithm used in a commercial treatment planning system. Currently, the initial Hessian fed into the algorithm is diagonal. The influence of including more accurate curvature information, represented as off-diagonal elements, is explored for three patient cases.
A more accurate initial Hessian results in a much faster progress of optimization than when using a diagonal initial Hessian. Furthermore, the optimal beam profiles differ significantly, with an accurate Hessian they are very jagged compared to the smooth profiles obtained with a diagonal Hessian. Jagged profiles are, in general, not desirable since they are harder to deliver, but for a certain class of IMRT problems they are preferable. The results also indicate that the IMRT problem is an ill-posed inverse problem in the sense that very different fluence profiles can produce almost identical dose distributions.

A Method for Approximating Symmetrically Reciprocal Matrices by Transitive Matrices

Dahl, Geir *(Center of Mathematics for Applications, University of Oslo, Norway)*

The problem of approximating symmetrically reciprocal matrices by transitive matrices has received some attention recently. This problem has applications in multicriteria decision theory. Several approximation approaches have been suggested and analyzed. We here suggest another approach, called the multiplicative approach. We show that the optimal approximation in this sense may be found efficiently by transforming the problem into a known combinatorial optimization problem (the minimum cycle mean problem) for which efficient and simple combinatorial algorithms exist.

Keywords: Transitive matrix, symmetrically reciprocal matrix, approximation.

Cutting Plane Method in Decision Analysis

Ding, Xiaosong *(Mid-Sweden University, Sweden)*
Al-Khayyal, Faiz *(Mid-Sweden University, Sweden)*

Computational decision analysis methods, such as the DELTA method, have been developed and implemented over a number of years for solving decision problems where vague and numerically imprecise information prevails. However, the evaluation phases in those methods often give rise to bilinear programming problems, which are time-consuming to solve in an interactive environment with general nonlinear programming solvers. This paper proposes a linear programming based algorithm that combines a cutting plane method with the lower bounding technique for solving this type of problem. The central theme is to identify the global optimum as early as possible in order to avoid generating unnecessary cuts in the convergent cutting plane procedure.

Topology Optimization of the Navier-Stokes Equations

Evgrafov, Anton *(Chalmers University of Technology, Sweden)*

We consider the problem of optimal design of flow domains for Navier-Stokes flows in order to minimize a given performance functional. We attack the problem using topology optimization techniques, or control in coefficients, which are widely known in structural optimization of solid structures for their flexibility, generality, and yet ease of use and integration with existing FEM software. Topology optimization rapidly finds its way into other areas of optimal design, yet until recently it has not been applied to problems in fluid mechanics. The success of topology optimization methods for the minimal drag design of domains for Stokes fluids (see the study of Borrvall and Petersson [Internat. J. Numer. Methods Fluids, vol. 41, no. 1 pp. 77-107, 2003]) has lead to attempts to use the same optimization model for designing domains for incompressible Navier-Stokes flows.
We show that the optimal control problem obtained as a result of such a straightforward generalization is ill-posed, at least if attacked by the direct method of calculus of variations. We illustrate the two key difficulties with simple numerical examples and propose changes in the optimization model that allow us to overcome these difficulties. Namely, to deal with impenetrable inner walls that may appear in the flow domain we slightly relax the incompressibility constraint as typically done in penalty methods for solving the incompressible Navier-Stokes equations.
In addition, to prevent discontinuous changes in the flow due to very small impenetrable parts of the domain that may disappear, we consider so-called filtered designs, that has become a "classic" tool in the topology optimization toolbox. Technically, however, our use of filters differs significantly from their use in the structural optimization problems in solid mechanics, owing to the very unlike design parametrizations in the two models.We rigorously establish the well-posedness of the proposed model and then discuss related computational issues.

A New Generating Set Search Method for Unconstrained Optimisation

Frimannslund, Lennart (*University of Bergen, Norway*)
Steihaug, Trond (*University of Bergen, Norway*)

Generating set searches, a class of derivative-free optimisation methods, has been an area of active research in recent years, much caused by the development of convergence theory. However, although these methods are usually easy to implement, robust and provably convergent in most cases, their attractiveness suffers from the fact that they are slow when it comes to convergence. Usually these methods do not take the local topography of the objective function into account.
We present a new algorithm which is a modification to a well known generating set search method, Compass Search. The new algorithm tries to adapt its search directions to the local topography by accumulating curvature information about the objective function as the search progresses. We present some theory regarding its properties, as well as numerical results that show our algorithm to outperform Compass Search most of the time, sometimes by significant relative margins, on noisy as well as smooth problems. In addition, preliminary numerical results indicate that we can exploit the sparsity information of the Hessian matrix. Thus allowing us to solve relatively large problems using methods in this class.

Tabu Search Heuristics for the Probabilistic Dial-a-Ride Problem

Ho, Sin C. *(University of Bergen, Norway)*
Haugland, Dag *(University of Bergen, Norway)*

We present an efficient neighborhood search procedure for the probabilistic dial-a-ride problem. The suggested approach requires O(n^4) computations as opposed to O(n^6) operations required by a straightforward neighborhood evaluation. In the current work a tabu search and a hybrid GRASP/tabu search exploiting this search procedure are developed and compared through numerical experiments.

Path Relinking for the Vehicle Routing Problem

Ho, Sin C. *(University of Bergen, Norway)*
Gendreau, Michel *(Université de Montréal, Canada)*

The aim of this work is to propose a tabu search heuristic with path relinking to solve the classical vehicle routing problem. Computational results show that using path relinking periodically in the search speeds up the search to find good solutions. They also show that tabu search with path relinking is able to produce better solutions than pure tabu search using much less computing time.

Stronger Necessary Conditions for the Existence of a Compatible OSPF Metric

Broström, Peter *(Linköping Institute of Technology, Sweden)*
Holmberg, Kaj *(Linköping Institute of Technology, Sweden)*

This presentation is a continuation of the presentation "Determining the Non-Existence of a Compatible OSPF Metric". It addresses the question of whether or not for a set of desired traffic patterns in an Internet Protocol telecommunication network using OSPF (Open Shortest Path First), there exists a compatible metric, i.e. weights making the routers give the specified traffic patterns. In the previous presentation it was shown that the existence of what we here call 1-valid cycles prove the non-existence of a compatible metric. In a 1-valid cycle the flow of two commodities is changed in a cycle. We here prove that a 2-valid cycle, which is a cycle in which more than two commodities are changed, exists if and only if there exists a 1-valid cycle. Furthermore, a 3-valid set of cycles is defined as a set of cycles where the flow of one commodity is changed in each cycle. Unfortunately we have not been able to show that the non-existence of 3-valid sets of cycles is sufficient for the existence of a compatible metric. However, for some special cases, such as when the desired traffic patterns only consist of a number of trees, stronger results are obtainable. Since it is fairly easy to find 1-valid cycles, we also consider the case when we know that there does not exist any 1-valid cycle.
An alternate title of this talk is "In Search of Sufficient Conditions for the Existence of a Compatible OSPF Metric". We can formulate sufficient conditions for the existence of a compatible metric, but at the moment this formulation is not practically usable. However, this talk aims to show that the gap between the necessary and sufficient conditions is decreasing.

Optimizing the Schedule of a Sports League

Joborn, Martin *(Carmen Systems AB, Sweden)*
Optimizing the game schedule of a sports league is a very complex problem, known as the traveling tournament problem. In a real situation, the problem includes many intangible constraints that are hard to quantify. Also, the objective function is quite fuzzy. In this presentation, we will compare the "theoretical" traveling tournament problem with a real instance. Further, we will sketch how the problem is solved today, discuss potentials for optimization, and outline how we have helped a major sports league to optimize their planning.

Ship Scheduling with Visit Separation Constraints

Sigurd, Mikkel M. *(University of Copenhagen, Denmark)*
Ulstein, Nina L. *(Norwegian University of Science and Technology, Norway)*
Nygreen, Bjørn *(Norwegian University of Science and Technology, Norway)*
Ryan, David M. *(University of Auckland, New Zealand)*

This talk discusses an application of planning support in designing a sea-transport system. Increased pressure on the road network and increasing transport needs make companies look for new transport solutions. This spurred an initiative to create a new liner shipping service. The initiative came from a group of Norwegian companies who need transport between locations on the Norwegian coastline and between Norway and The European Union. While few producers on the Norwegian coast have sufficient load to support a cost efficient, high frequency sea-transport service, they can reduce costs and decrease transport lead-time by combining their loads on common ships. They agreed upon a tender (transport offer) which was proposed to a number of shipping

companies. The tender specifies the number of cargos per week and time constraints for pickup and delivery. It also states the requirements regarding ship types and loading and unloading techniques. For rapid handling, all goods must be transported in containers. Finally the tender specifies the yearly payment each company will make to be part of this transportation system. Today there are neither ships nor harbour facilities to support the proposed solution. Thus, major investments are necessary. Estimates indicate that investments in ships alone, can amount to about 150 mill US dollars. We present a model which calculates a near optimal fleet and corresponding routes to satisfy the requirements in the tender. The problem is a variant of the general pickup and delivery problem with multiple time windows. In addition, it includes requirements for recurring visits, separation between visits and limits on transport lead-time. The problem is formulated as a set partitioning model and solved by a heuristic branch-and-price algorithm.

CPLEX Overview and Recent Advances in Mathematical Programming

Oussedik, Sofiane *(ILOG, France)*

The first part of the presentation is an overview of ILOG CPLEX algorithms and parameters for solving linear, mixed integer, quadratic and mixed integer quadratic programs. Also, CPLEX includes the ILOG CPLEX Callable Library (C and VB APIs) and ILOG Concert Technology (C++, Java and .Net APIs) to make it easy to embed the powerful CPLEX algorithms in your application. The second part of the presentation will highlight the recent algorithmic advances in Mathematical programming and the features that made CPLEX the industry standard.

Duality in MIP

Pachkova, Elena V. *(Copenhagen University, Denmark)*

This presentation treats duality in Mixed Integer Programming (MIP in short). A dual of a MIP problem includes a dual price function *F*, that plays the same role as the dual variables in Linear Programming (LP in the following).
The price function is generated while solving the primal problem. However, different to the LP dual variables, the characteristics of the dual price function depend on the algorithmic approach used to solve the MIP problem. Thus, the cutting plane approach provides nondecreasing and superadditive price functions while branch and bound algorithm generates piecewise linear, nondecreasing and convex price functions.
Here a hybrid algorithm based on branch and cut is investigated, and a price function for that algorithm is established. This price function presents a generalization of the dual price functions obtained by either the cutting plane or the branch and bound method.

Global Optimality Conditions for Discrete and Nonconvex Optimization, With Applications to Lagrangian Heuristics and Column Generation

Larsson, Torbjörn *(Linköping University, Sweden)*
Patriksson, Michael *(Chalmers University of Technology, Sweden)*

The well-known and established global optimality conditions based on the Lagrangian formulation of an optimization problem are consistent if and only if the duality gap is zero. We develop a set of global optimality conditions that are structurally similar but are consistent for any size of the duality gap. This system characterizes a primal-dual optimal solution by means of primal and dual feasibility, primal Lagrangian epsilon-optimality, and, in the presence of inequality constraints, delta-complementarity, that is, a relaxed complementarity condition. The total size epsilon + delta of those two perturbations equals the size of the duality gap at an optimal solution. The characterization is further equivalent to a near-saddle point condition which generalizes the classic saddle point characterization of a primal-dual optimal solution in convex programming.
The system developed can be used to explain, to a large degree,when and why Lagrangian heuristics for discrete optimization are successful in reaching near-optimal solutions. Further, experiments on a set covering problem illustrate how the new optimality conditions can be utilized as a foundation for the construction of Lagrangian heuristics. Finally, we outline possible uses of the optimality conditions in column generation algorithms and in

the construction of core problems, and illustrate our findings on instances of the generalized assignment problem.

Origin-Destination Matrx Estimation from Traffic Counts

Peterson, Anders *(Linköping University, Sweden)*

Origin--destination matrices, which specify the travel demand for all pairs of origin and destination nodes in a traffic network, can be estimated by utilizing link flow observations. We will describe and motivate an optimization model formulation of the generic problem and discuss how it can be handled with respect to an implicit problem, modelling the route choice mechanism. Special attendance will be given to the problems occurred by introducing time-dependence to the model.

Sensitivity Analysis of a Bilevel Traffic Equilibrium Problem with Welfare Constraints

Rydergren, Clas *(Linköping University, Sweden)*

Transport planners currently face a major challenge to devise future transport plans to meet multiple expectations and objectives. In this research, we aim to develop a decision-support tool for enhancing the understanding of various transport policies and finding appropriate transport measures. We are developing a suitable model for the urban transport system, together with flexible mathematical forms for expressing efficiency, equity and public acceptability considerations in the form of objectives and constraints. The model is intended to be used for studying the impact of various policies based on the use of sensitivity analysis expressions of the inputs to the model. In this presentation a bilevel model is given together with solution methods for the lower level problem and the corresponding sensitivity analysis problem.

Facility Location under Economics of Scale in the Case of Uncertain Demand

Schütz, Peter *(University of Science and Technology, Norway)*
Stougie, Leen *(University of Science and Technology, Norway)*
Tomasgard, Asgeir *(University of Science and Technology, Norway)*

The presentation adresses facility location under uncertain demand. The problem is to determine the optimal location of facilities and allocation of customer demand to these facilities. The costs of operating the facilities are subject to economics of scale and customer demand is uncertain. The objective is then to minimize the total expected cost. These costs can be split into three parts: firstly the costs of investing in a facility and maintaining it, secondly the costs of operating a facility with strictly diminishing average costs, and thirdly linear transportation costs. We show a solution method for this problem based on Lagrangean Relaxation. We present computational results from the Norwegian meat industry and the location of slaughterhouses.

A Stochastic Algorithm for Constrained Multiobjective Optimization

Shukla, P. K. *(Indian Institute of Technology Kanpur, India)*

A stochastic method is presented for solving constrained multiobjective optimization problems. This method may be thought of as an extension of Schäffler's method (which is based on solution of stochastic differential equation) for the solution of unconstrained multiobjective problems. Several methods for constraint handling are presented in this paper. Numerical results on several test problems are given. Problems with a large number of variables as well as with complex search space can be handled by this method. Finally using the above stochastic method an algorithm for constrained global multiobjective optimization is presented.

Modified Variable Neighborhood Search for the Vehicle Routing Problem with Accessibility Constraints

Souid, Mahdi *(UVHC/LAMIH/ROI, France)*
Hanafi, Saïd *(UVHC/LAMIH/ROI, France)*
Semet, Fréderic *(UVHC/LAMIH/ROI, France)*

The classical capacitated Vehicle Routing Problem (VRP) consists in determining optimal delivery routes for a set of homogeneous vehicles to serve a set of customers. Each route is covered by one vehicle without exceeding its capacity. Moreover, each route starts and ends at the same depot. Each customer is served exactly once. In this paper, we consider the Vehicle Routing Problem with Accessibility constraints (VRPA) which is defined on a graph of which vertices are partitioned into two sub-sets V1 and V2, served by two types of vehicles, i.e. Truck and truck + trailer. The customers of V1 are accessible by both vehicles types whereas the customers of V2 are only accessible by the trucks. The VRPA is a generalization of the VRP, it possesses numerous applications in domains such as logistics, economic planning of distribution networks and their management.
The classic capacitated vehicle routing problem, a special case of the VRPA where V2 is empty, has been studied extensively. The VRP is known to be NP-Hard, so VRPA is also a NP-hard problem. Generally, exact methods for NP-hard problem do not allow even moderately-sized problems to be solved. Heuristic approaches are needed to solve large scale instances of practical problems. Variable Neighborhood Search (VNS), introduced by Hansen and N. Mladenovic is a recent metaheuristic which exploits systematically the idea of neighbourhood change, both in the descent to local minima and in the escape from the valleys which contain them. For solving the VRPA by VNS we exploit the connection of this location and routing problem with close and particular cases. The neighborhood structures used can be classified in three categories according to the number of routes involved in the corresponding move : i) for a unique route we use the generalized adding/dropping procedure as proposed in GENIUS heuristic for traveling salesman problem; ii) for the two routes we use classical VRP moves such that dropping, adding, swapping; iii) for several routes we consider the move which consist to open or close a depot as done in location problem.
We propose various implementations of a Modified Variable Neighborhood Search (MVNS) method for the resolution of the VRPA, differentiated basing on the following criteria: local search method, choice of the neighbor solution, Sequence of neighborhoods. Test problems were generated in order to validate and determine the best implementation. MVNS method gives good results for VRPA. An improvement of MVNS method can be obtained by hybridization with a tabu search method.

The Hub Location Network Design Problem

Thomadsen, Tommy *(Technical University of Denmark)*
Stidsen, Thomas *(Technical University of Denmark)*

Designing hierarchical telecommunication networks pose some very difficult optimization problems. Most solutions today involve sequential solution of a series of easier optimization problems. In this presentation we will present the Hub Location Network Design (HLND) problem. The HLND problem combines the routing problem, the network design problem and the hub selection problem into one problem. The objective is to minimize the link establishment costs and the link capacity costs. We present an ILP model for the HNLD problem. To solve larger instances of the problem we develop a cut-and-price algorithm for the LP problem, which includes additional cuts to tighten the gap. Based on the LP solution IP solutions are generated. In most cases the gap is zero.
The hub location network design problem is related to several wellknown optimization problems: Network design, Generalized Travelling Salesman and Location-Routing. The connection between the HLND problem and these will briefly be discussed.

Design of Planar Articulated Mechanisms Using Branch and Bound

Stolpe, Mathias *(Technical University of Denmark)*
Kawamoto, Atsushi *(Technical University of Denmark)*

In this talk we present an optimization model and a solution method for optimal design of two-dimensional mechanical mechanisms. The mechanism design problem is modeled as a nonconvex mixed integer program

which allows the optimal topology and geometry of the mechanism to be determined simultaneously. The underlying mechanical analysis model is based on a truss (pin jointed assembly of straight bars) representation allowing for large displacements. For mechanisms undergoing large displacement elastic stability is of major concern. We derive conditions, modeled by nonlinear matrix inequalities, that guarantee that a stable mechanism is found. The feasible set of the design problem is described by nonlinear constraints as well as nonlinear matrix inequalities.
To solve the mechanism design problem a branch and bound method based on convex relaxations is developed. The relaxations are strengthened by adding valid inequalities to the feasible set. Encouraging computational results, which will be presented, indicate that the branch and bound method can reliably solve mechanism design problems of realistic size to global optimality.

Joint Hub Location, Node Clustering and Network Design of Two-Tiered Meshed Networks

Thomadsen, Tommy *(Technical University of Denmark)*
Stidsen, Thomas *(Technical University of Denmark)*

In this talk we discuss design of two-tiered meshed networks. A two-tiered meshed network consists of clusters of nodes comprising the access network tier and a backbone tier which interconnects the clusters. Each cluster contains exactly one hub node which routes the traffic between clusters.
Designing a two-tiered meshed network involves a number of interrelated problems: Hub location, clustering of nodes and network design. These problems have often been carried out independently, but since the problems are interrelated, this may lead to suboptimal designs. We determine hub location, clustering of nodes and network design jointly. A mathematical model is presented for the problem and a bound is derived. Also a GRASP heuristic is implemented to obtain feasible solutions.
Tiers exists because of limitations in communication equipment, e.g. hop limits, organizational advantages, e.g. easier upgrade and the observation, that a two-tiered network seems to cope with changes in the traffic better than a network without tiers. However, enforcing tiers does incur some additional cost. This is clear, since any two-tiered network is also a feasible solution when networks without tiers are considered. For that reason we investigate how much cost is incurred by enforcing two tiers, i.e. we compare with networks without tiers.

Supply Base Management

Wallace, Stein W. *(Molde University College, Norway)*
Aas, Bjørnar *(Molde University College, Norway)*

The purpose of this presentation is to outline a rather new project at Molde University College. The project is in cooperation with Statoil, the major Norwegian oil company, and is focused on the supply base Vestbase in the town of Kristiansund, north of Molde. The base supplies about ten drilling and production platforms off the Norwegian coast with all types of equipment they need for daily operations. Supply vessels are used for goods and helicopters for people. Our main focus is on the scheduling of vessels to the platforms.

Methods for Some Linear and Quadratic Optimization Problems Defined on a Set of Orthogonal Vectors

Wedin, Per-Åke *(Umeå University, Sweden)*

For several practical optimization problems one wants to find a minmizer that belongs to a set of orthonormal vectors. In most cases these problems are 3-dimensional and related to rigid body movement, tereophotogrammetry and similar applications. However,there are also, e.g. in psychometrics, quadratic or linear optimization problems over a set of m*n-matrices Q with orthonormal columns a.k.a. a Stiefel manifold. Interesting properties that set problems of this kind apart are the following: (1) The function to be minimized is nice. It is always convex, while the set that we optimize over is non-convex and fairly tricky. (2) It is possible to get a useful unconstrained LOCAL representation of the optimization problem while the constrained representation is needed globally. (3) Some optimization problems of this kind, e.g. the Procrustes problem, have

a unique minimum thar can be attained by using the singular value decomposition. (4) There are several optimization problems of this kind where the number of local minima will grow very fast with the number of orthogonal vectors to be optimized over. (5) Geometrical considerations combined with Lagrange multipier theory are useful analytic tools.
Thomas Viklands, Umeå, has developed an algorithm and furthered the analysis for one problem of this kind, the Penrose-Procrustes regression problem. Viklands has shown that this problem can have 2^n local minima. Here n is the number of orthogonal vectors for which the optimization problem is defined. Vikland´s algorithm tries to find all the minima of the P-P problem.
In this talk we will summarize the state of the art of the research in this area with special emphasis on the Penrose-Procrustes regression problem.

A Hierarchical Neighbourhood Search Method for Topology Optimization

Svanberg, Krister *(Royal Institute of Technology, Sweden)*
Werme, Mats *(Royal Institute of Technology, Sweden)*

In topology optimization of continuum structures, a fixed design domain is given. The infinite dimensional problem then deals with finding an optimal subdomain of the given design domain to fill with material. In practice, the design domain is discretized, so that the objective and constraint functions can be computed via the finite element method. The design of the structure is represented by binary design variables indicating material or void in the various finite elements.
We present a hierarchical neighbourhood search method for solving topology optimization problems defined on discretized linearly elastic continuum structures. Two different designs are called neighbours if they differ in only one single element, in which one of them has material while the other has void. The proposed neighbourhood search method repeatedly jumps to the ``best'' neighbour of the current design until a local optimum has been found, where no further improvement can be made. The ``engine'' of the method is an efficient exploitation of the fact that if only one element is changed (from material to void or from void to material) then the new global stiffness matrix is just a low rank modification of the old one. To further speed up the process, the method is implemented in a hierarchical way. Starting from a coarse finite element mesh, the neighbourhood search is repeatedly applied on finer and finer meshes. Numerical results are presented for minimum weight problems with constraints on respectively the stiffness of the structure, strain energy densities in all non-void elements, and von Mises stresses in all non-void elements.

Minimum-Energy Broadcasting and Multicasting in Ad Hoc Networks: Some Integer Programming Formulations and Computational Experiences

Yuan, Di *(Linköping University, Sweden)*

Broadcast (multicast) routing in a wireless network involves the construction of a broadcast (multicast) tree used by a source node to send messages to some other nodes in the network. The energy consumption of the tree is the sum of the transmission power at the nodes. The optimization problem of finding a broadcast (multicast) tree of a minimum amount of energy arises in applications of wireless networking where network units must be energy-aware. An example of such wireless systems is ad hoc networks. In this talk we present some integer programming formulations for this problem and report our computational experiences.
