3rd Student Conference on Operational Research

SCOR 2012, April 20–22, 2012, Nottingham, UK

Edited by

Stefan Ravizza
Penny Holborn

OASICS

Editors

Stefan Ravizza	Penny Holborn
School of Computer Science	School of Mathematics
University of Nottingham	Cardiff University
`smr@cs.nott.ac.uk`	`HolbornPL@cardiff.ac.uk`

ACM Classification 1998
G.1.6 Optimization, G.2.1 Combinatorics, G.2.2 Graph Theory, G.2.3 Applications, I.2 Artificial Intelligence, I.6 Simulation and Modeling

ISBN 978-3-939897-39-2

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at http://www.dagstuhl.de/dagpub/978-3-939897-39-2.

Publication date
June, 2012

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Digital Object Identifier: 10.4230/OASIcs.SCOR.2012.i

ISBN 978-3-939897-39-2 **ISSN 2190-6807** **http://www.dagstuhl.de/oasics**

OASIcs – OpenAccess Series in Informatics

OASIcs aims at a suitable publication venue to publish peer-reviewed collections of papers emerging from a scientific event. OASIcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

ISSN 2190-6807

www.dagstuhl.de/oasics

Contents

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn
OASICS OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Preface

We are delighted to present the proceedings of the 3rd Student Conference on Operational Research (SCOR 2012). The aim of SCOR is to provide PhD students in the early stages of their OR careers with an excellent opportunity to meet other researchers with similar interests and to present their work in a relaxed and friendly environment. In its third edition the event was bigger and more international than ever before.

The conference took place from the 20–22 April 2012 at the University of Nottingham and welcomed PhD students, mainly from European Universities, studying Operational Research, Management Science or a related field. Research areas included: Data Mining, Decision Support, Forecasting, Graphs and Networks, Healthcare, Heuristics and Metaheuristics, Inventory, Mathematical Programming, Multicriteria Decision Analysis, Neural Networks and Machine Learning, Optimisation, Reliability and Risk Assessment, Scheduling and Timetabling, Stochastic Modelling, Supply Chain Management, Simulation and System Dynamics and Transport.

Some statistics from the event are as follows: there were 88 participants from 18 countries who provided 72 talks in up to 4 parallel streams over 3 days. There were 4 plenary talks given by Gavin Blackett (The OR Society), David Buxton (DSE Consulting Ltd.), Tony O'Connor (GORS) and Vincent Knight (Cardiff University) together with Louise Orpin (The OR Society). These talks were extremely beneficial as they were aimed at giving delegates a feel for what could be next in their OR careers outside of academia.

In total there were 6 sponsors and we are extremely grateful for the sponsorship obtained, especially from The OR Society and from the ASAP research group at the University of Nottingham, which made this event possible.

As far as we are aware, this was the first time that a Smartphone app was used at an OR event. This complemented the programme and book of abstracts. The main advantage of this app was that participants were able to personalise their own schedule for the event.

The review process was based on the presentation, quality and originality of the research and there were at least two referees assigned to each paper. In total 21 submissions were received with 11 papers successfully accepted. These were from 6 different countries including Germany, Hungary, Italy, Poland, Serbia and the United Kingdom demonstrating the international presence at the event.

We would like to give a special thanks to all authors who submitted a paper for review, to the Committee for their continued support and to all who contributed to the success of SCOR 2012 and the proceedings.

Stefan Ravizza
Penny Holborn

Organisation

Conference Committee

- Stefan Ravizza, University of Nottingham, UK (Chairman)
- Penny Holborn, Cardiff University, UK (Vice-Chair)
- Michael Clark, University of Nottingham, UK
- Emily Cookson, Lancaster University, UK
- Magdalena Gajdosz, University of Strathclyde, UK
- Pablo Gonzalez Brevis, University of Edinburgh, UK
- Izabela Komenda, Cardiff University, UK
- Urszula Neuman, University of Nottingham, UK
- Martin Takáč, University of Edinburgh, UK
- Alessia Violin, Université Libre de Bruxelles, Belgium

Additional Reviewer

- Timo Kunz, Lancaster University, UK

Sponsors

- The OR Society
- ASAP research group, University of Nottingham
- Prospect Recruitment
- Gower Optimal Algorithms Ltd.
- Tata Steel
- Banxia Software

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A Case Study on Optimizing Toll Enforcements on Motorways*

Ralf Borndörfer, Guillaume Sagnol, and Elmar Swarat

Zuse Institute Berlin Department Optimization
Takustr. 7, 14195 Berlin, Germany
`{borndoerfer,sagnol,swarat}@zib.de`

Abstract

In this paper we present the problem of computing optimal tours of toll inspectors on German motorways. This problem is a special type of vehicle routing problem and builds up an integrated model, consisting of a tour planning and a duty rostering part. The tours should guarantee a network-wide control whose intensity is proportional to given spatial and time dependent traffic distributions. We model this using a space-time network and formulate the associated optimization problem by an integer program (IP). Since sequential approaches fail, we integrated the assignment of crews to the tours in our model. In this process all duties of a crew member must fit in a feasible roster. It is modeled as a Multi-Commodity Flow Problem in a directed acyclic graph, where specific paths correspond to feasible rosters for one month. We present computational results in a case-study on a German subnetwork which documents the practicability of our approach.

1998 ACM Subject Classification G.1.6 Optimization, G.2.3 Applications

Keywords and phrases Vehicle Routing Problem, Duty Rostering, Integer Programming, Operations Research

Digital Object Identifier 10.4230/OASIcs.SCOR.2012.1

1 Introduction

The Vehicle Routing Problem (VRP) is an extensively studied optimization problem with a lot of variants and very different solution approaches, see [8, 4] for an overview. The core is always to determine a set of tours to execute given tasks. In this paper we will present a model to set up tours as well, but under some unusual settings and assumptions that lead to another variant of vehicle routing problems.

We address the problem of computing tours for toll control inspectors on motorways. In 2005 Germany introduced a distance-based toll on motorways for commercial trucks with a weight of at least 12 tonnes. The enforcement of the toll is the responsibility of the German Federal Office for Goods Transport (BAG). It is implemented by a combination of an automatic enforcement by stationary control gantries and by random tours of mobile control teams. There are about 300 control teams distributed over the entire network. The teams consist mostly of two inspectors, but in some cases of only one. Each team can only control highway sections in its associated *control area*, close to the depot. Germany is subdivided into 21 control areas. Our approach could also be applied to other countries and toll systems if they use mobile control tours. Furthermore there must be central databases that provide on-demand information on drivers, if they have paid tolls or not.

* This work was funded by the German Federal Office for Goods Transport (BAG).

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn; pp. 1–10
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Our challenge is to solve the VRP for the mobile teams. The tours should guarantee a network-wide control whose intensity is proportional to given spatial and time dependent traffic distributions. Similar to classical vehicle routing problems we have a *length restriction* for all tours according to daily working time limitations. We model this problem using a space-time network and formulate an associated optimization problem as an Integer Program. A typical problem instance is to compute a monthly plan for one control area of the German network.

The paper is structured as follows. In Section 2 the *Toll Enforcement Problem (TEP)* [2] is introduced and distinguished from classical vehicle routing approaches. In the following section the graph and IP model for the TEP is introduced and the integration of the rostering part is presented. In Section 4 we explain our settings for the case study and the computational experiments. Finally, in Section 5 the results are discussed and some directions for future research are provided.

2 Optimal Toll Enforcement

In contrast to most of the Vehicle Routing problems, where a set of given demands or tasks has to be met, in the TEP this is different. Since the number of teams is fixed, the goal here is to control as efficiently as possible with the available personnel. If we assign a profit value to each section that could be covered by a tour, then our problem is related to a *vehicle routing problem with profits* or a *prize-collecting vehicle routing problem*. In the case of only one vehicle this is known as the prize-collecting TSP (or TSP with profits). There are only a few applications for the case of several vehicles, see Feillet et al. [5] for a literature survey. A suitable approach to prize collecting in our setting is to set the profit to the number of trucks that pass through a motorway section during a predefined time interval. This has the effect to reward the controls on highly utilized sections. Furthermore, the profit values differ during different time intervals. For example, the section with the highest profit might not be the same during the rush hour and during the night. Hence, not only the sections of a tour must be determined, but also the starting time and the duration of a section control.

A second difference is with respect to driver assignments. Naturally vehicle routing problems result in a set of tours. Drivers are assigned to the tours in a subsequent step. The feasibility of crew assignments is not part of the algorithms to solve the classical models. But in the toll control setting it is not possible to ignore the availability of crews. There are only a few drivers that can perform a planned tour since each tour must start and end at the home depot of its associated team. Thus, sequential approaches to plan the tours independently of the crews will fail.

If we assign a crew to each tour, it must fit within a feasible crew roster, respecting all legal rules, over a time horizon of several weeks. Minimum rest times, maximal amounts of consecutive working days, and labor time regulations must be satisfied. Hence, a personalized duty roster planning must be used in our application. Therefore, we developed a novel integrated approach, that leads to a new type of vehicle routing problems. To the best knowledge of the authors there is no optimization approach to toll enforcement in the literature yet. Related publications deal with problems such as tax evasion or ticket evasion in public transport; they mainly discuss the expected behavior of evaders or payers from a theoretical point of view, e.g. [1], or optimal levels of inspection, see [3].

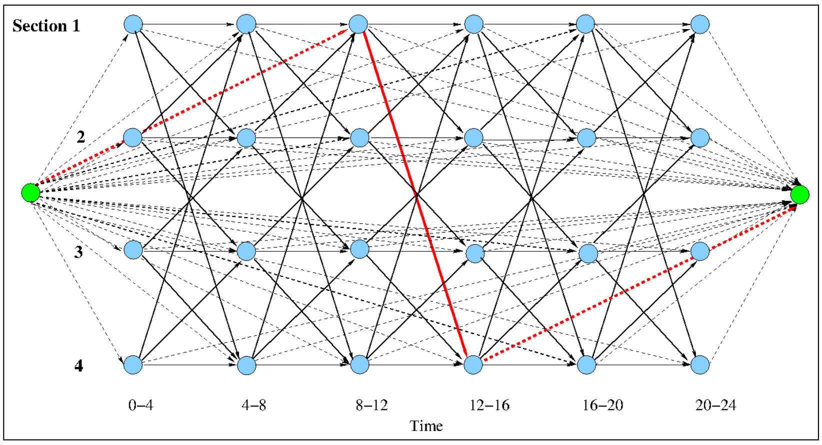


Figure 1 Construction of the tour planning graph.

3 A Graph Model for the Planning of Inspector Tours

The TEP can be described in terms of a *section graph* $G = (S, N)$. The nodes $s \in S$ correspond to so-called *control sections*, that is, sub-parts of the network with a length of approximately 50 km. The edges $n \in N$ of G connect two section nodes, if they have at least one motorway junction or motorway exit in common. The temporal dimension of our model involves a given planning horizon T, e.g., one month, and some time discretization Δ, e.g., two hours. According to the time discretization, we extend G to a space-time digraph $D = (V, A)$, the *tour planning graph*. Its nodes $v \in V$ are either defined as a pair of a section and a point in time, i.e., $v = (s, t) \in S \times [0, T]$, or they represent start and end nodes d_s and d_t for the vehicle paths (depot nodes). Directed arcs connect either adjacent time intervals for the same section, i.e., $a = ((s, t_1), (s, t_2))$ with $t_2 = t_1 + \Delta$, starting at $t_1 = 0$ until $t_2 = T$, or they connect adjacent sections, i.e., $(s_1, s_2) \in N$ implies $((s_1, t_i), (s_2, t_{i+1})) \in A \, \forall t_i \in \{0, \Delta, \ldots, T - \Delta\}$. In addition, arcs that connect the start depot node with all other non-depot nodes and all non-depot nodes with the end depot node model the beginning and the end of a tour. Figure 1 illustrates this construction by a network with four sections and a time discretization of $\Delta = 4$ hours. One drawback of this approach is that the size of D depends on Δ. With $\Delta = 4$ a feasible tour consists in controlling two sections, e.g., see the red path in Figure 1. In practice, there is always a break for the drivers after the first half of the sections in the tour is controlled. Hence, the time and location of the break does not need to be modeled explicitly in D.

A profit value and a length is associated with each arc $a \in \delta^+(v), v \in V$ to collect the profit for visiting v during a control tour. We consider the problem of finding a (d_s, d_t)-path in D for each vehicle f on each day that respects the restriction of a maximum tour length. This is called the *Tour Planning Problem (TPP)*. We model the TPP as a classical 0/1 Multi-commodity flow problem [7] in D.

Let P be set of all paths in D, that represent feasible control tours and $P_{f,j} \subset P$ the set of all paths that are feasible for vehicle $f \in F$ and start at day $j \in J$. In addition for a section $s \in S$, the set of all paths $p \in P$ that visit a node $v = (s, t_i) \in V$ is denoted by P_s. By κ_s the minimum control quota, i.e., the minimum number of control visits on section s during the planning horizon, is indicated. We introduce binary variables z_p, $p \in P$, to decide that a tour is chosen or not. Then the following IP solves the TPP:

$$\max \sum_{p \in P} w_p z_p \tag{1}$$

$$\sum_{p \in P_{f,j}} z_p \leq 1, \qquad \forall (f,j) \in F \times J \tag{2}$$

$$\sum_{p \in P_s} z_p \geq \kappa_s, \qquad \forall s \in S \tag{3}$$

$$z_p \in \{0,1\}, \quad \forall p \in P. \tag{4}$$

In the objective function (1) the profit of the selected tours is maximized. Constraints (2) guarantee that each vehicle performs at most one tour per day. Constraints (3) requires that at least κ_s paths, that traverse section s, are chosen in a feasible solution. Finally constraints (4) demand the path variables being binary.

3.1 Integration of Duty Roster Planning

The second task in the TEP is the planning of the rosters, called the *Inspector Rostering Problem (IRP)*. There, the objective is to minimize the total costs. In contrast to many other duty scheduling and rostering approaches the goal in this setting is not to minimize crew costs. In the IRP the costs penalize some feasible but inappropriate sequences of duties, see Section 4 for examples. This is more related to the criterion of driver friendliness.

We formulate the IRP again as a Multi-Commodity flow problem in a directed graph $\tilde{D} = (\tilde{V} = (\hat{V} \cup \{s,t\}), \tilde{A})$ with two artificial start and end nodes s, t. The nodes $\hat{v} \in \hat{V}$ represent duties as a pair of day and time interval. The arcs $(u, \hat{v}) \in \tilde{A} \subseteq \tilde{V} \times \tilde{V}$ model a feasible sequence of two duties according to legal rules.

Let M be the set of all inspectors and b_m the start value of the time account of m. For each month there is a regular working time for each inspector. This needs not to be met exactly, but there is a feasible interval for the nominal value of the working time account. Hence, by l_m we denote the lower bound for the nominal value of inspector m and by u_m the upper bound, respectively. In addition, let t_v be the duration of duty $v \in \hat{V}$. The costs of a direct sequence of duties u and $v \in \hat{V}$ in a roster are indicated by $c_{u,v}$. A variable $x^m_{u,v}$ for each arc (u,v) and inspector m is introduced. According to that we propose the following integer programming formulation for the IRP:

$$\min \sum_{m \in M} \sum_{(u,v) \in \tilde{A}} c_{u,v} x^m_{u,v} \tag{5}$$

$$\sum_{v} x^m_{s,v} = 1, \qquad \forall m \in M \tag{6}$$

$$\sum_{k} x^m_{v,k} - \sum_{u} x^m_{u,v} = 0, \qquad \forall v \in \hat{V}, m \in M \tag{7}$$

$$b_m + \sum_{u \in \hat{V}} \sum_{v} t_u x^m_{u,v} \leq u_m, \qquad \forall m \in M \tag{8}$$

$$b_m + \sum_{u \in \hat{V}} \sum_{v} t_u x^m_{u,v} \geq l_m, \qquad \forall m \in M \tag{9}$$

$$x^m_{u,v} \in \{0,1\}, \quad \forall (u,v) \in \tilde{A}, m \in M. \tag{10}$$

As already mentioned, in the objective function (5) the cost is minimized. By Constraints (6) we assure that exactly one arc per inspector with a non-zero flow value is leaving depot s. The resulting path of all non-zero flow arcs for an inspector is called *Inspector Roster Path*. The flow conservation in the non-depot nodes is expressed by constraints (7). The inequalities (8) and (9) enforce for each inspector that the planned roster does not exceed the interval for the nominal value of the working time account. In the last constraint (10) the flow variables are restricted to be binary. In this model the use of arcs variables allows to handle small and medium size instances, i.e., instances that have up to 160000 flow variables in the rostering part.

Finally a formulation for the TEP is derived, by combining the TPP and the IRP by so-called *coupling constraints*. To this end, by $P_{f,v}$ we define the set of all control paths feasible for vehicle f and duty $v \in \hat{V}$. In addition, the parameter n_f gives the number of inspectors in vehicle f and $m \in f$ denotes, that inspector m uses vehicle f in a fixed assignment. This leads to the following equation:

$$\sum_{p \in P_{f,u}} n_f z_p - \sum_{m \in f} \sum_{v} x^m_{u,v} = 0 \quad \forall f \in F, u \in \hat{V} \tag{11}$$

Each control path p belongs to a predefined time interval. Hence, by (11) it is guaranteed that for each control path p in D all inspectors in the corresponding team have a feasible roster path, where a duty in the time horizon of p is scheduled.

The objective function of the TEP is therefore a combination of collecting the profit (1) and minimizing the cost (5). In practice, we maximize a linear combination of these two objectives. A coefficient is used to set the proportion of the rostering costs in the integrated model. We have observed that for several instances, the solution which maximizes the profit (1) contains no penalized duty sequence arcs (i.e., (5) is at its minimum). More details about the rostering costs will be discussed in the next sections.

4 Case Study – Instances and Settings

We have implemented the above described model in an optimization tool, called TC-OPT. We tested TC-OPT on some real world instances from one control area with about 20 inspectors. We used a set of standard (legal) rules, like minimum rest times, working time regulations and some other constraints mentioned in the sections above. In addition, manually generated reference plans from this control area are given. This allows a comparison of our novel approach with plans that are representative for the current manual planning of the control tours.

We selected six instances, three for August 2011 (aug1, aug2, aug3) and three for October 2011 (oct1, oct2, oct3). Table 1 distinguishes the instances according to several criteria. The column "mincontrol for all sections" indicates whether we used the minimum control quota constraint, see eq. (3), for all sections (case "yes") or if some sections can be omitted during control (case "no"). The fourth column gives information about so called "rotation penalties" used in our model. This relates to the artificial costs we introduced in Section 3.1. A sequence of two duties d_1 and d_2 of an inspector from day t to day $t+1$ is called a *rotation*, if the starting-time of d_2 is different from d_1. If it starts later, e.g., from Mo 8-17 to Tu 10-19, we call this a *forward rotation*. If the second duty begins earlier, e.g., Mo 8-17 to Tu 6-15, it is a *backward rotation*.

Table 1 Overview on general settings for the test instances. All other parameter and data, e.g., inspectors, teams, sections or holidays, are the same for all instances. If the data depends on the selected month, it is the same in all instances belonging to the same month.

instance	Δ	mincontrol for all sections	rotation penalty	traffic data from
aug1	4	yes	moderate	last month
aug2	2	yes	moderate	last month
aug3	4	no	moderate	last month
oct1	4	yes	moderate	last month
oct2	4	yes	strong	last month
oct3	4	yes	moderate	last year

It is legal to use rotations in a duty roster, if the minimum rest time between the end of a duty and the beginning of the next is not less then 11h. Beside of that, is it an important goal to avoid or to minimize rotations in a roster. Because it is more employee-friendly if subsequent duties start always on the same time and if changing to another start time occurs only after some days off. According to that we integrated *rotation (penalty) factors* in our model which must be chosen in relation to the profit of the control tours. The value “strong” means that the penalty factors are higher then the profits of all tours while for “moderate” this holds only for a majority of the tours but not for all. In general the factor of the backward rotation should be higher, since this strongly infects the length of rest times between duties. The last column indicates the period, from where we took the profit values in the objective function of the TPP (1). All data depending on the selected month, like holidays, fixed duties or working time accounts, are same in all of the three instances belonging to the same month. All other data, e.g. team assignments or the selection of sections, are the same for all instances.

Another important aspect of the control planning is that a control may not start at any time. There are given time intervals when the tours can take place. We call them *working time windows*. For our test setting we used six different time windows, two starting in the morning, one mid-day interval and three that start in the afternoon or in the evening. A major constraint in our model is that a certain duty mix is maintained. Therefore, for each time window there is a minimal and maximal contingent of all duties, e. g., the duties from 6am to 3pm must be at least 20% of all duties and at most 50%. The main significance of these constraints is to define upper bounds on the number of late and night duties.

5 Case Study - Results and Discussion

We were able to solve all instances with a proven optimality. There is no optimality gap with more than 10%. Hence, for all instances we received a feasible control plan. Before the solution behavior is dicussed in detail in Section 5.1 the quality of the optimized plans is analysed first. Comparing the manual and the optimized plan we see several benefits in using TC-OPT. It is easier to handle the balance of the working hours, see eq. (8) and (9), especially in the case of different working hours in a team. The second is that we could comply with the duty mix constraints, which is more difficult in manual planning. Furthermore, it was possible to prove the benefit in introducing the rotation factors. Comparing instance oct2 to oct1 we were able to reduce the number of rotation when increasing the factor. For some instances, e.g., aug1, even a low factor suffices to avoid rotations between two scheduled duties.

In Figure 2 the distribution of the controls on the 15 sections of the control area is shown.

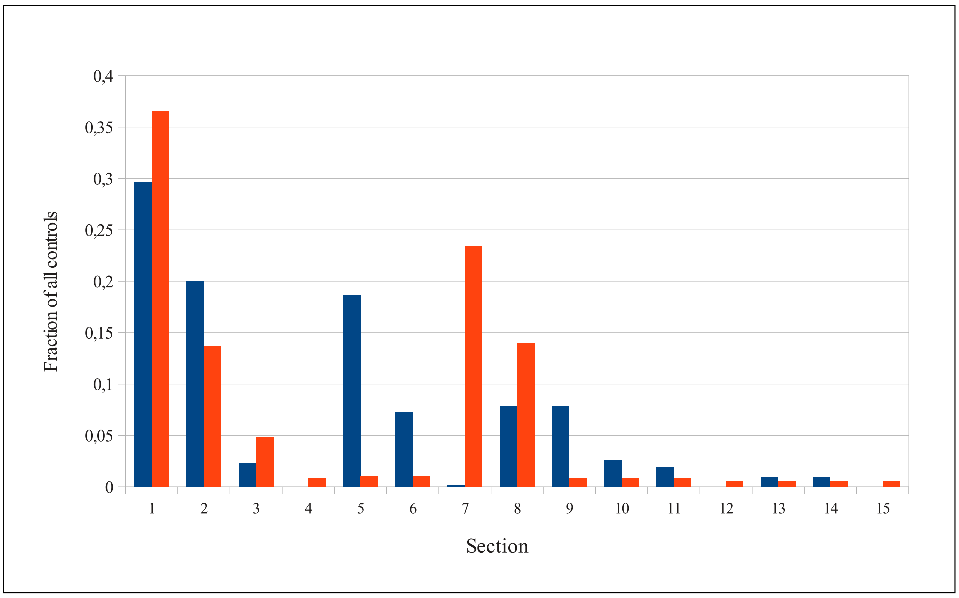


Figure 2 Control distribution over all sections in percent for instance aug1, the left blue columns show the reference plan, while the right columns in red represent the TC-OPT solution.

The main difference between the optimized and the reference plan is that the latter controls a lot on sections five and six, while the optimized plan has a focus on sections seven and eight. The difference originates from the observation, that there is significantly more traffic on sections seven and eight than on five and six. The same holds for sections one and two, where TC-OPT controls the first section twice as much as the second one. So the objective function clearly tends towards the sections with the most traffic. The sections with very low traffic, like 12 or 15, were only controlled by the required minimum quota. We can conclude, that the control is mostly planned according to the traffic distribution. The use of minimum control quota constraints (3) achieves a better control coverage of the whole network compared to the reference plans.

Choosing the time discretization to two hours, as in aug2, the control distribution along the sections is quite similar to the four-hour case. The only difference is a slightly higher part of the control on some low traffic sections. This originates from the possibility to control up to four sections during a tour in the two-hour case. As a result, one can control high traffic sections as well as low traffic sections in a common tour.

Another important aspect is the control distribution according to different days in a week. The main focus of the control is between Monday and Friday according to the fact that there is much less traffic at the weekend. One reason for that is the Sunday truck ban on German motorways that tolerates only small exceptions, e.g., for some urgent food transports.

Beside of the distribution over the weekdays it is interesting to study the distribution during a day, i.e., a daily control pattern. It is important to mention that those values heavily depend on the chosen duty mix constraints. Usually labour agreements restrict the number of late and night shifts. Those regulations have a major impact on the mix constraints and thereby also on the daily control pattern. According to that, our optimization tool allows the planners to predefine the intervals for the duty mix constraints. This may have the side effect that the distribution can differ a lot between different areas.

Figure 3 shows a comparison of the optimized plan (in red) and the reference plan (in

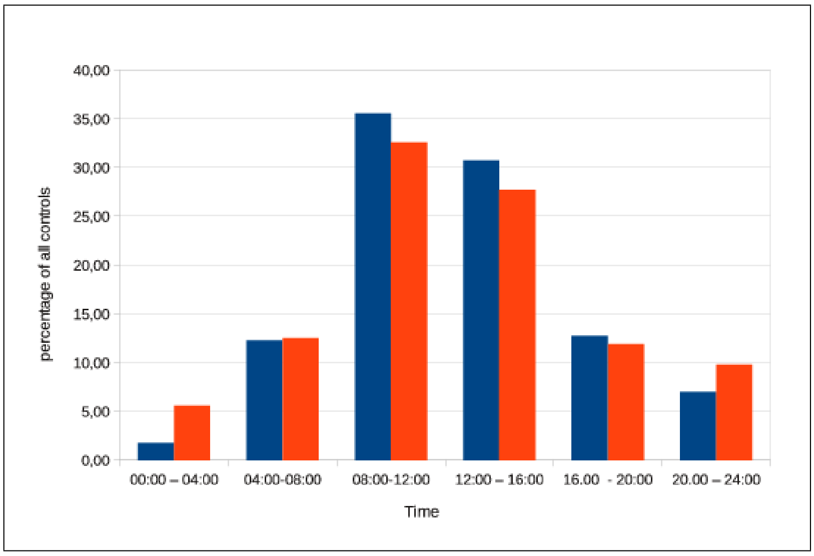


Figure 3 Control distribution of instance aug1 for different time intervals across a day in percent. The blue columns show the reference plan, while the red represent a solution by TC-OPT.

blue) according to different time intervals across a day. The values are summarized over all days of the planning period and depend on the exemplary chosen duty mix setting of our test. As one can see, the part of control during late evening and night is higher in the TC-OPT result. This follows from the observation that on some motorway sections, in particular on central network axis, the truck traffic during the night is not much less comparing to the traffic during the day. Consequently, the profit value for a control on some highly utilized sections at night is higher than the value for a control on a low traffic section during the day. Hence, TC-OPT tries to schedule more night duties.

5.1 IP Solution Analysis

After discussing the quality of the solution regarding to several important aspects in practice, we analyse the solution behavior of our algorithmic approach. All computations were done on a Blade Server with an 8-core Intel Xeon CPU with 3.2 GHz and SUSE Linux 11.4 as operating system. The memory limit for the solution tree was 40 GB. Furthermore, there was a time limit of 2 days (= 172800 seconds) for each instance. As an IP Solver CPLEX 12.3 [6] by IBM with the default parameter setting was applied by using up to eight threads. Table 2 presents all relevant data for the solution analysis. The second and third column display the number of rows and columns of the IPs. As mentioned before, the chosen time discretization has a huge influence of the size of the IP, especially on the number of columns. The next column shows the root LP value, v(lp), i.e., the value of the linear relaxation of the IPs. This value is the first dual bound value during the solution process. Since our problem is a maximization problem, the dual bound value is an upper bound on the optimal solution. The fifth column gives the dual bound value b^* at the end of the solution process. In the sixth column the best primal bound v^* is denoted, which is the best integer value or in other words the best solution found. The next columns display the gap between primal and dual bound, the time, when the first solution was found and the overall solution time,

Table 2 IP-Solution analysis for all test set instances. The value of the root LP is denoted by v(lp) and the best integer solution by v^*. By b^* we name the value of the dual bound at the end of the optimization run. The time limit for the optimization run equals $2days = 48h = 172800sec$. The column "time 1st sol." indicates when the first primal solution was found while "time(ip)" gives the overall solution time.

in-stance	#rows	#co-lumns	v(lp)	b^*	v^*	gap(%)	time 1st sol.[sec.]	time(ip) [sec.]
aug1	11424	198179	569790.98	524510.24	523232.20	0.24	3900	172800.00
aug2	14574	771893	562176.71	519017.06	515664.10	0.65	14700	172800.00
aug3	11424	198179	570545.77	526971.61	525560.03	0.27	2400	172800.00
oct1	11799	211680	615269.50	570377.87	535365.49	6.54	6240	170474.0[1]
oct2	11799	211680	615269.50	570517.72	520215.84	9.67	6050	139514.0[1]
oct3	11799	211680	582188.86	542402.84	511003.29	6.14	7900	147333.3[1]

i.e., time(ip). The gap is computed by $(b^* - v^*)/v^*$.

The most important result is that we were able to compute feasible solutions for all instances with a gap of at least 10%. The gap for all August-instances was even less than 1% at all, even for the huge 2-hour-discretization instance. This encouraging result is enhanced by the fact that for each instance a first feasible solution could be found during the first two and a half hours of the solution time. In addition, the initial gap, i.e., the gap after the first solution, is less then 9% for all august-Instances and less than 21% for all october-instances. An interesting observation is that the instance with the biggest number of rows and columns, aug2, is not the one with the highest solution gap. All October-instances were much more difficult to solve than the August-instances. The reason for this lies in the duty mix constraints that were much more tighter in the October-instances. This makes it more difficult to find feasible solutions that satisfy the small feasibility intervals for the time windows.

Beside of that the October-instances lead to an huge Branch & Bound tree that even exceeds the very high amount of memory. On the one hand, this is a point, where our models should be improved in the future, but on the other hand the wasteful memory consumption relates partly to the before-mentioned duty mix requirement that is chosen too strict.

5.2 Conclusion and Future Research

We presented the first model of optimizing the tours of toll control inspectors on motorways. This problem was derived as a special Vehicle Routing Problem with profits where also the crew scheduling has to be taken into account. Therefore, our model consists of two parts. The first one, the Tour Planning Problem, is to plan tours of inspectors in the network. The second part, the Inspector Rostering Problem, builds up a feasible duty roster of each inspector. For both problems an appropriate graph model was presented. We formulated the two as Multi-Commodity Flow Problems in their planning graphs. Each of the problem is formulated by an IP, and by coupling constraints they are integrated into one common formulation. The model is implemented in a tool, named TC-OPT.

The main issue of this paper was the presentation of a case-study on the TEP. We optimized several duty plans of two different planning horizons for one exemplary chosen

[1] Memory-Limit reached

control area. A set of standard legal rules was integrated in our model. We were able to solve all six instances with only a small optimality gap within two days. Furthermore, a comparison with a reference plan, that represents different aspects of traditional planning approaches, showed that TC-OPT could support the planners. It was able to improve the quality of the optimized plan in many ways. Another important aspect of our case study is the duration of the solution process. According to the usual planning horizon a duty plan has to be computed only once in a month. Hence, our time limit of two days is reasonable from a practical point of view. In urgent cases the time limit can be significantly reduced since the first solution is found after a few hours in most cases.

Hence, it can be concluded that we are on the right way to get an optimization tool that satisfies all requirements of the toll control planning problem. An important aspect in our future research is to be able to compute feasible plans for all control areas, with different settings in a reasonable time by moderate hardware requirements. We will test some impacts in our model to get smaller computation times, like problem-dependent reduction techniques on the sizes of the planning graphs or variations on several parameters and strategies of the MIP solution process. Also additional rules, that are not legally defined, but very common in practice, will be integrated in our model. A typical example is a fairer planning of late and night shifts.

References

1. M.G. Allingham and A. Sandmo. Income Tax Evasion: a Theoretical Analysis. *Journal of Public Economics*, 1(3-4):323–338, 1972.
2. R. Borndörfer, G. Sagnol, and E. Swarat. An IP Approach to Toll Enforcement Optimization on German Motorways. Technical Report ZIB Report 11-42, Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany, 2011.
3. C. Boyd, C. Martini, J. Rickard, and A. Russell. Fare Evasion and Non-Compliance: a Simple Model. *Journal of Transport Economics and Policy*, 23(2):189–197, 1989.
4. J.-F. Cordeau, G. Laporte, M. W.P. Savelsbergh, and D. Vigo. Vehicle Routing. In C. Barnhart and G. Laporte, editors, *Handbooks in Operations Research and Management Science, Transportation*, volume 14, chapter 6, pages 367–428. Elsevier, 2007.
5. D. Feillet, P. Dejax, and M. Gendreau. Traveling Salesman Problems with Profits. *Transportation Science*, 39(2):188–205, 2005.
6. IBM ILOG CPLEX, IBM Software Group. *User-Manual CPLEX 12.3*, 2011.
7. R. M. Karp. On the Computational Complexity of Combinatorial Problems. *Networks*, 5(1):45–68, 1975.
8. P. Toth and D. Vigo. *The Vehicle Routing Problem.* Society for Industrial and Applied Mathematics, Philadelphia, USA, 2002.

Revenue maximization through dynamic pricing under unknown market behaviour

Sergio Morales-Enciso[1] and Jürgen Branke[2]

1 **Centre for Complexity Science, The University of Warwick**
Coventry, CV4 7AL, UK
S.Morales-Enciso@warwick.ac.uk

2 **Warwick Business School, The University of Warwick**
Coventry, CV4 7AL, UK
Juergen.Branke@wbs.ac.uk

Abstract

We consider the scenario of a multimodal memoryless market to sell one product, where a customer's probability to actually buy the product depends on the price. We would like to set the price for each customer in a way that maximizes our overall revenue. In this case, an exploration vs. exploitation problem arises. If we explore customer responses to different prices, we get a pretty good idea of what customers are willing to pay. On the other hand, this comes at the cost of losing a customer (when we set the price too high) or selling the product too cheap (when we set the price too low). The goal is to infer the true underlying probability curve as a function of the price (market behaviour) while maximizing the revenue at the same time. This paper focuses on learning the underlying market characteristics with as few data samples as possible by exploiting the knowledge gained from both exploring potentially profitable areas with high uncertainty and optimizing the trade-off between knowledge gained and revenue exploitation. The response variable being binary by nature, classification methods such as logistic regression and Gaussian processes are explored. Two new policies adapted to non parametric inference models are presented, one based on the efficient global optimization (EGO) algorithm and the second based on a dynamic programming approach. Series of simulations of the evolution of the proposed model are finally presented to summarize the achieved performance of the policies.

1998 ACM Subject Classification I.2.6 Learning

Keywords and phrases Dynamic pricing, revenue management, EGO, Gaussian processes for classification

Digital Object Identifier 10.4230/OASIcs.SCOR.2012.11

1 Introduction and literature review

Dynamic pricing is a strategy which aims to offer different prices for the exact same product to different customers. In general, this strategy is followed in order to maximize a firm's revenue by understanding how the market reacts to different prices.

Depending on the nature of the service or product offered, there are different possible scenarios. When designing a dynamic pricing policy, the distinction on whether finite or infinite inventories and time horizons are being considered is important. Another possible distinction is the nature of the market in terms of buying recurrence and the existence of a precedent reference price because recurrent customers for an already established product have been shown to develop a peak-end memory effect which influences their behaviour towards price changes [13]. In this paper, we consider the memoryless scenario with an infinite time horizon and infinite inventory. This is commonly the case when a new non seasonal product

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn; pp. 11–20
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

is introduced to the market and both the expected life cycle of the product and the available inventory are sufficiently large.

Dynamic pricing strategies have been long studied in the context of physical distribution channels where advertised prices are targeted to the whole market and every price change carries a cost. Examples of this can be tracked back to revenue management research in the airline industry started in the 70s [10], or to [7] where optimal policies are studied for seasonal products with finite inventories sold through physical distribution channels. These studies try to understand the behaviour of the demand in terms of the arrival rates, which is affected after a period of time when prices are changed for a given epoch. Since all the market has access to the same information (advertised price), a change in the arrival rate of customers who actually buy is expected.

In the current market conditions, the internet offers a perfect scenario for dynamic pricing, since prices can be changed individually for each customer without incurring any cost. In fact, this is already a common practice among internet retailers as is shown by the controversial example of Amazon.com, which in September of 2000 ran a randomized pricing test across different customers [17], along with many other examples which are described in [9]. Many recent studies considering the internet as a distribution channel and accounting for more frequent price changes have been developed. An interesting characteristic of the recent studies is that they all consider the relation between the advertised price and the arrival rate of the buying customers as a descriptive measure of the market behaviour, like when dealing with physical distribution channels. Furthermore, most of the reviewed papers use a parametric regression model when inferring the demand curve, imposing a functional form to the unknown market behaviour (e.g. [3], [1], and [4]). Some of the studies justify the choice of a parametric model (like [6]), and only a few make use of non parametric models [2].

In order to take full advantage of the virtual markets, and keeping in mind that the firm aims to maximize the accumulated revenue and not to control the market behaviour, different prices can be quoted to each customer without disclosing the quoted price to the rest of the market. However, the fact that the information is not available to everyone removes the relationships between the quoted prices and the rate of arriving customers. For instance, reducing the price of a product will not necessarily increase the arrival rate of customers unless it is widely advertised, which is exactly what would be avoided in real cases if the market price sensitivity is to be studied. Because of this, we propose to estimate the overall market price sensitivity through the estimation of the probability of buying a product given a quoted price (see section 2), and we assume constant arrival rates, removing the need of a time index. The traditional approach to this problem and our proposed approach are equivalent in the sense that the arrival rate of customers with probability 1 of buying –as modelled in the former– and the probability of customers (buying and not buying) arriving at a constant rate –as proposed it the latter– are interchangeable. The difference mainly relies in the number of samples needed to understand the market behaviour and the way of performing the experiment, i.e. quoting undisclosed prices directly to the customer.

The goal of this paper is to provide insight on which is the best pricing policy to follow in order to maximize the accumulated revenue of a firm having to determine the price of a product in an unknown market under the described framework. We propose 2 policies (EGO and one step lookahead in revenue) and compare their performance with another 2 policies known in literature (random exploration and greedy) as well as with the optimal solution as a benchmark, since in a realistic scenario it would be unknown. EGO (Effective global optimization) policy is based on the methodology outlined in [8] and takes samples at the maximum expected improvement point. The one step lookahead policy is based on a dynamic programming approach and maximizes the overall revenue of the 2 next samples, which implicitly takes into account the information gained during the first sample, but does not include an explicit term to quantify information acquisition as opposed to the one step lookahead policy proposed in [6].

The next section addresses the problem of how to infer the probability of getting a positive answer from a customer given a quoted price, which is required by the policies to function. Section 3 details the derivation of the compared policies taking care of the mathematical details, and section 4 describes the implementation details and the results obtained for each of the compared policies. Finally, in section 5 our conclusions and future research paths are presented.

The main contributions of this paper are first, the use of Gaussian process for classification (GPC), a non parametric method of inference, as detailed in section 2. Second, the design of two new policies, which are well suited to work with non parametric models. Third, the consideration of multimodal markets as explained in 2, and –more importantly– fourth, the change of paradigm to approach the problem by proposing to use the probability of buying rather than the arrival rates as a description of the market behaviour.

2 Market behaviour inference

Every time a price is quoted to a customer, the customer has the choice to accept the product at the quoted price or to reject it. If the probability distribution for a customer accepting an offer at every given price were known it would be straightforward to determine the price to be quoted so that it would maximize the expected income. Nevertheless in our case, and often in real life, the probability distribution of obtaining a positive answer from the customer is not known for every possible price. This could be determined by making an extensive survey, but it would be suboptimal since many samples would be required at very low and very high prices which do not provide any profit. Besides, it is desirable to start maximizing the profit from the first quotes.

The aim in this section is to determine an accurate probability distribution $\mathbb{P}(y = 1|x; \mathcal{D})$ for each price $x \in [0, x_{max}]$ given the minimum possible number of observations $\mathcal{D} = \{(x_i, y_i)_{i=1}^n\} = (X, Y)$ so that the optimal price to be quoted to the customer can be found. This means that the response variable $y \in \{0, 1\}$ is binary by nature.

2.1 Logistic regression

One possibility is to use logistic regression, which assumes a functional form (1) for the market demand, and despite the name is a generalized linear model which works as a k-class classifier rather than as a regression. As shown in chapter 4 of [5], Bayesian logistic regression allows to express not only the expected mean probability, but also confidence in the estimate by using some approximations of which the interested reader can find further details in [12]. If the market is composed of only one type of customer, a logistic regression of first order should be used. But if for example the market is composed of more than one type of customer, each with different price sensitivity, the aggregated distribution would be multimodal and the inference process would require the inclusion of higher order transformations in the design matrix Φ in (1) in order to accurately capture this property. Multimodal markets are common and as a simple example we can consider a university, where staff members might be willing to pay more than students for a same product. A more realistic example is an online retailer selling products across different countries with different incomes and preferences.

$$\mathbb{P}(y=1|x;\mathcal{D}) = \frac{1}{1+e^{-\omega^T\Phi(X)}} \text{ , where } \omega^T \text{ is the transpose of the coefficient vector } \omega. \quad (1)$$

Knowing the number of modes beforehand will be a problem if the composition of the market is unknown. Failing to use the correct degree for the regression will result in a poor estimate, which limits in a considerable fashion the performance of any sampling policy using this inference method when the composition of the market is unknown. This is the main reason why logistic regression is not incorporated in any of the proposed sampling policies. Another drawback of the logistic regression for our goal is that it is difficult to incorporate prior information. In the Bayesian framework, the prior information relates to the coefficients ω to be inferred, and provides a way to express a posterior distribution on the coefficients inferred from the data. But for design matrices of order higher than 1, the functional form of the resulting curve is not trivial to control through ω.

2.2 Gaussian processes for classification (GPC)

In order to avoid the problem of determining the possibly multimodal composition of the market and to allow more flexibility to the model to adapt to the true shape of the market behaviour, a non parametric model is suggested. In particular, a GPC is used.

When used for regression, a Gaussian process (GP) is fully defined by a mean function which allows to introduce any prior information available into the model, and a covariance function which expresses the correlation between the data points [14]. As a result of applying GP for regression to a dataset, we obtain an estimate on the function generating the data, also called latent function f, along with the confidence of such estimate. This means that not only do we get the best fit of a function to the data, but also a distribution for each point of the function expressing how certain we are about the obtained estimate. Since the range of the latent function is $\mathbb{R}$, f is not suitable to be interpreted as a probability. So, in order to ensure the output falls in the interval [0,1], which is required for the classification case, a sigmoid function (λ) is applied to f. A complete and formal description on GP can be found in [14] and [11].

For our case, the GPC is defined by the following mean (m), covariance (k), and sigmoid (λ) functions:

$$m(x) = 0, \quad k(x,x') = \sigma_f exp\left(-\frac{(x-x')^2}{2l^2}\right), \quad \lambda(f) = \frac{1}{1+e^{-f}} \qquad (2)$$

Using a zero mean function means that no prior information is being introduced, or equivalently, the probability of a customer buying the product is $\frac{1}{2}$ a priori. Nevertheless, introducing any prior information on the shape of the resulting probability curve to be inferred would be as simple as changing the prior mean to the believed shape in order to improve the inference process. Furthermore, if the probability curve were suspected to follow a monotonously decreasing behaviour with respect to price, a fair assumption in many cases, it could be done by following the proposed methodology in [15]. The squared exponential covariance function (2) specifies how much a given data point influences the points in its vicinity and how far the vicinity extends to. The optimal parameters $\theta^* = (\sigma_f^*, l^*)$ are to be learnt from the available observations by maximizing the logarithm of the likelihood of the parameters given the data ($log(\mathcal{L}(\theta|\mathcal{D})) = -\frac{1}{2}Y^TK^{-1}Y - log|K| - \frac{n}{2}log(2\pi)$) with respect to θ. K is the $n \times n$ covariance matrix containing the resulting value of the kernel function of all possible combinations of the observations. $\lambda(f)$ (2) is the logistic function which is used to shrink

f so that the output can be interpreted as a probability. Once the optimal parameters are known, the expected value of f evaluated at a given price is estimated using (3) and the variance of the estimate is given by (4).

$$\mathbb{E}[f(x_{new})|\mathcal{D}, \theta^*] = k(x_{new}, X)K^{-1}Y \tag{3}$$

$$Var[f(x_{new})|\mathcal{D}, \theta^*] = k(x_{new}, x_{new}) - k(x_{new}, X)K^{-1}k(X, x_{new}) \tag{4}$$

Where $k(x_{new}, X)$ is the $1 \times n$ row vector resulting from applying the kernel function from the new data point to all the data points in $\mathcal{D}$ and $k(X, x_{new}) = k(x_{new}, X)^T$.

Finally, the probability of a customer accepting the quote given the price x_{new} is given by (5) and the confidence of this prediction is given by (6).

$$\mu(x) := \mathbb{P}(y_{new} = 1|x_{new}; \mathcal{D}) = \lambda\Big(\mathbb{E}[f(x_{new})|\mathcal{D}, \theta^*]\Big) \tag{5}$$

$$\sigma^2(x) := Var[\mathbb{P}(y_{new} = 1|x_{new}; \mathcal{D})] = \lambda\Big(\mathbb{E}[f(x_{new})|\mathcal{D}, \theta^*] + Var[f(x_{new})|\mathcal{D}, \theta^*]\Big) - \lambda\Big(\mathbb{E}[f(x_{new})|\mathcal{D}, \theta^*]\Big) \tag{6}$$

Figure 1 shows the resulting probability of buying $\mathbb{P}(y = 1)$ for all the prices as inferred from $n = 5$ observations using a GPC. The observations were obtained by sampling from a bimodal market, which is represented by the weighted sum of the probabilities of buying which are Bernoulli distributed with parameter $q(x)$ as shown in plot (a) in Figure 1.

3 Policies to compare

The purpose of a policy is to have a rule which dictates the best action to take given our current knowledge at any given point in time, providing a systematic way of taking decisions. Under stochastic conditions an optimal policy can only guarantee up to some probability that the recommended action will be the best. In general, when a process can be simulated, repeated realizations of the same process are performed in order to learn the best course of action. When trying to learn the market behaviour, there are two main challenges. First, the goal is to learn the transition probabilities governing the process, i.e. the distribution of the possible outcomes given the current state and action taken. This makes the process impossible to simulate, since the probability distribution underlying the samples is unknown. And second, the value of taking an action, i.e. the obtained revenue for a given quote in this case, depends not only on the current state but on all the history of actions taken on which the current belief is based, making the process non-Markovian. This makes the problem to be a partially observable non Markov decision process (POnMDP), which can be treated as a POMDP where the state space grows exponentially each time an action is taken [16].

This paper considers five policies. The first, called optimal policy or Π_{true}, is provided only as a benchmark and upper bound for the others since it requires the true probability curve to be known. It consists simply of repeatedly quoting the price x^* that maximizes the expected revenue given complete information. The next two are standard policies that have been proposed earlier (c.f. [6] for example) out of which one is the random policy Π_r and the other is the greedy policy Π_g. Π_r takes samples at the price x resulting from a uniform distribution over the whole interval of possible prices $x^* \sim U[x|0, x_{max}]$. This means

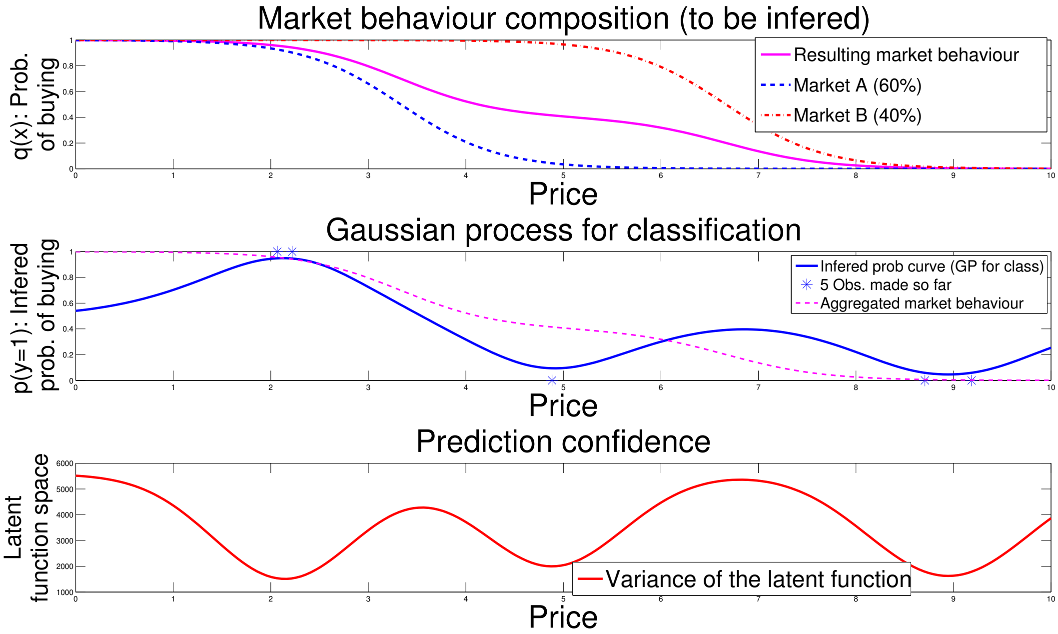


Figure 1 Illustration of market behaviour inference using GPC. In (a), a bimodal aggregated market behaviour resulting from the composition of two types of customers with different proportions is shown. The estimated probability of a customer accepting a quote resulting from using a GPC with only 5 observations is illustrated in (b), and (c) shows the variance of our estimations made in (b) which decreases (i.e. confidence increases) where there are samples to support the belief.

that it only focuses on exploration in a completely uninformed way and does not focus on exploitation at all. Finally two new policies are proposed and explained in detail: EGO and one step lookahead in revenue.

3.1 Greedy policy Π_g

The greedy policy seeks to maximize the immediate revenue, so it samples where the expected revenue at the current time is maximized given the most recent state of knowledge $\mathcal{D}_n$. The expected revenue $\mathbb{E}_{x_n}[R(x_n)|\mathcal{D}_n]$ is calculated directly by multiplying the probability of a customer accepting a quote given the quote times the quoted price. The next best sample is chosen according to (7).

$$x_n^* = \underset{x_n}{\text{argmax}}(\underset{x_n}{\mathbb{E}}[R(x_n)|\mathcal{D}_n]) = \underset{x_n}{\text{argmax}}\left(x_n \mathbb{P}(y_n = 1|x_n, \mathcal{D}_n)\right) \quad (7)$$

3.2 EGO Policy Π_{ego}

EGO stands for efficient global optimization, and is a methodology proposed in [8], illustrated in particular in the engineering context where the need of fitting response surfaces from data samples often arises. The algorithm is based on sampling where the expected improvement is maximized. The improvement $I(x)$ is defined as the difference between the current maximum of the expected revenue known so far r^* and any other possible revenue at the given x provided it is larger than r^* ($I(x) = 0$ otherwise). To calculate the expected improvement $\mathbb{E}[I(x)]$, the probability distribution of the possible revenues at x must be known. This can be calculated since the GP provides the distribution of the belief across the possible

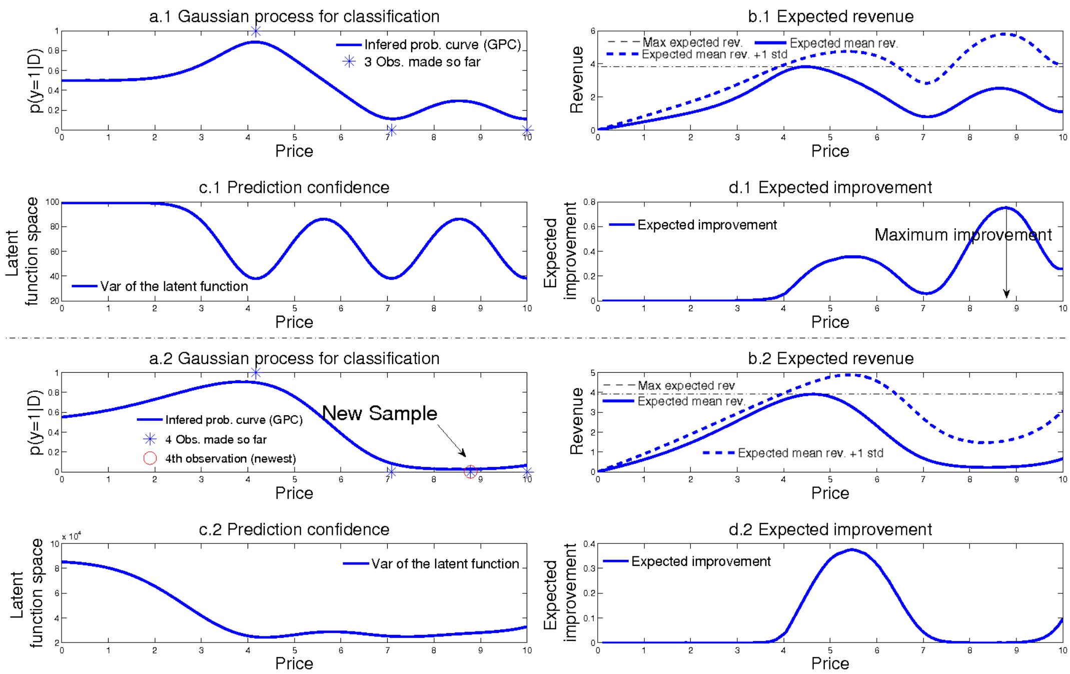


Figure 2 Two steps of the EGO policy in action are shown. Starting with (a.1), which shows the estimated probability of buying given the 3 shown data points, and (c.1) showing the confidence of the predictions, the expected revenue along with its confidence interval can be calculated as (b.1) illustrates. Then, using (9), the expected improvement can be calculated (d.1) and the best action to take determined by using (10).

values, which is specified by the mean (5) and variance (6). By multiplying these two values times the price x, we obtain the distribution on the expected revenue at x, which is used to compute the expected improvement (9).

$$I(x) = \max(r^* - \mathbb{E}[R(x)], 0) \quad \text{where} \quad r^* = \max_x(\mathbb{E}[R(x)]) \tag{8}$$

$$\mathbb{E}[I(x)] = \int_{r^*}^{\infty} I(x)\mathbb{P}(\mathbb{E}[R(x)])dx \quad \text{where} \quad \mathbb{P}(\mathbb{E}[R(x)]) \sim \mathcal{N}(\mathbb{E}[R(x)]|\mu(x), \sigma^2(x)) \tag{9}$$

Once the expected improvement $\mathbb{E}[I(x)]$ is known, the next best sample should be taken where $\mathbb{E}[I(x)]$ is maximized (10).

$$x^* = \underset{x}{\operatorname{argmax}}(\mathbb{E}[I(x)]) \tag{10}$$

The EGO policy explicitly takes into account both information acquisition and exploitation of what is believed to be the action with the highest reward. Besides, as the number of samples increases, the confidence intervals narrow, placing each time less weight in the exploration part. An illustration of how this policy works is provided in Figure 2.

3.3 One step lookahead in revenue Π_{dp1}

The one step lookahead in revenue policy proposes to sample at the maximum of the sum of the immediate expected revenue given the current observations $\mathbb{E}[R(x_n)|\mathcal{D}_n]$ plus the expected revenue at the next step given the current data together with the outcome of the action taken in the first step $\mathbb{E}[R(x_{n+1})|\mathcal{D}_n \cup \{(x_n, y_n)\}]$ appropriately weighted by the current belief (11). The first part of the sum corresponds to the greedy policy Πg. Since the action taken in step n influences our belief on the market behaviour, in order to calculate the second part of (11), all the possible outcomes of action x_n along with their possible responses y_n and the corresponding belief update, which follows from having a new sample, should be taken into account.

$$\mathop{\mathbb{E}}_{x_n}[R(x_n)+R(x_{n+1})] = \max_{x_n\in\mathcal{X}}\left\{\mathop{\mathbb{E}}_{x_n}\left[R(x_n)|\mathcal{D}_n\right]+\mathop{\mathbb{E}}_{x_n}\left[\max_{x_{n+1}}(\mathop{\mathbb{E}}_{x_{n+1}}[R(x_{n+1})|\mathcal{D}_n\cup\{(x_n,y_n)\}])\right]\right\} \tag{11}$$

Let $\mathbb{P}_n := \mathbb{P}_n(y_n = 1|\mathcal{D}_n)$ denote the estimated probability of a customer buying a product considering the observations available at step n, let $\mathbb{P}^+_{n+1} := \mathbb{P}_{n+1}(y_{n+1} = 1|\mathcal{D}_n \cup \{(x_n, y_n = 1)\})$ denote how the estimation of the probability would update at step $n + 1$ had the action x_n been taken and had its outcome been a positive answer $y_n = 1$, and let $\mathbb{P}^-_{n+1} := \mathbb{P}_{n+1}(y_{n+1} = 1|\mathcal{D}_n \cup \{(x_n, y_n = 0)\})$ denote the case where the outcome for action x_n had been a negative answer $y_n = 0$.

Since there are only two possible outcomes for y_n, and the revenue obtained when $y_n = 0$ is 0, only the cases with positive answers contribute to the expected revenue. However, the two possible outcomes must be taken into account after taking action x_n, since in both cases the belief would be updated in a different manner. Calculating the expectations given the available data, obtained by multiplying the price times the probability of getting a yes, we obtain:

$$\mathop{\mathbb{E}}_{x_n}[R(x_n)+R(x_{n+1})] = \max_{x_n}\left\{x_n\mathbb{P}_n+\mathbb{P}_n\max_{x_{n+1}}\left(x_{n+1}\mathbb{P}^+_{n+1}\right)+(1-\mathbb{P}_n)\max_{x_{n+1}}\left(x_{n+1}\mathbb{P}^-_{n+1}\right)\right\} \tag{12}$$

Finally, the next best sample according to Π_{dp1} is given by finding the price to be quoted such that it maximizes (12), which is expressed in (13)

$$x^*_n = \operatorname*{argmax}_{x_n}\mathop{\mathbb{E}}_{x_n}[R(x_n) + R(x_{n+1})] \tag{13}$$

4 Implementation and results

The accumulated revenue achieved throughout a given number of quotes is considered in order to compare the performance of the policies described above. The accumulated revenue is given by the sum of the products of the quotes made times the response obtained: $R_{acc} = X \cdot Y$.

To see how each of the policies perform, a simulation of the process was implemented and the accumulated revenue was tracked for the first 50 samples. The market is assumed to follow a bimodal probability curve $q(x)$ as shown in plot (a) in Figure 1. So, sampling the market is simulated by drawing a response from a Bernoulli distribution with parameter $q(x)$.

For each of the proposed policies, the simulation starts with 2 samples $\{x_0, x_1\} \in [0, 10]$ and their response $\{y_0, y_1\} \in \{0, 1\}$. If there is at least one positive and one negative response, then the policies start to run. Otherwise, new samples are taken until there is at least one of each possible responses. This is done to ensure the inference process is not mislead from the beginning. The new samples (before the policies start to run) are taken at $min(X)/2$ if Y is only composed of zeros, or at $max(X) + (10 - max(X))/2$ if Y is composed only of ones.

For Π_{true} and Π_r there is no need to perform any inference process. For the rest of the policies, each time a new sample is taken, the belief of the probability of buying is updated

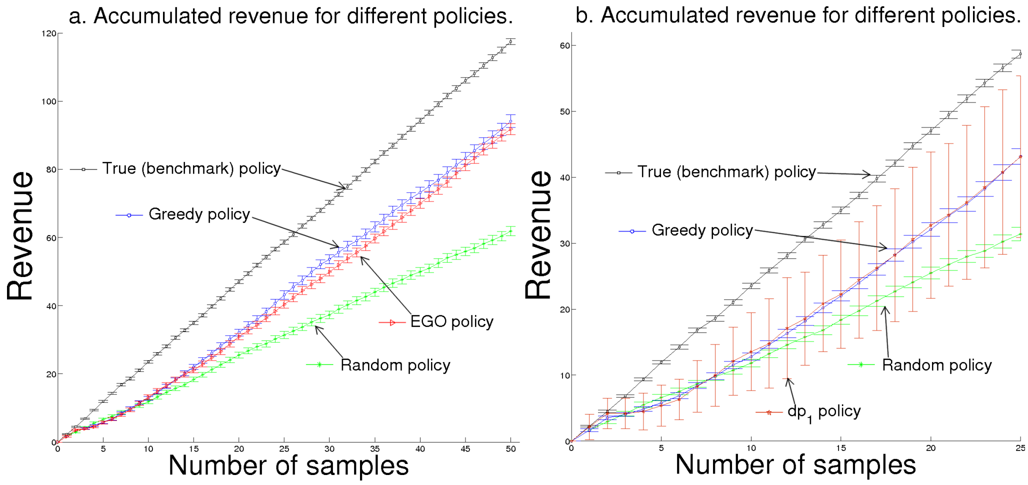


Figure 3 Performance of the five policies described in section 3. No statistically significant difference was found in the performance for policies Π_g, Π_{ego}, and Π_{dp1}, although they all shown better performance compared to Π_r.

by running the GPC and the price at which the next best sample is taken (x^*) is determined by applying each policy.

For all the policies except Π_{dp1} the simulation was run for 100 replications, each with different random seeds, but common random seeds were kept across different policies. The accumulated revenue for each policy and each replication was tracked along 50 samples, allowing to provide the results with confidence intervals. This is shown in Figure 3(a). Π_{dp1} was only ran for 50 replications and up to 25 samples because of its computational requirements. For clarity, the obtained results are presented in a separate plot in Figure 3(b).

After running the numerical simulations, it was found that the random policy Π_r performed the worst. This is due to the focus on constant uninformed exploration and the lack of exploitation. Nonetheless, the simulations show no statistical difference between the other 3 policies compared (Π_g, Π_{ego}, and Π_{dp1}) which seems counter intuitive. One possible explanation could be that no additional information is used across these 3 policies, even if the information available is being treated differently.

5 Conclusion and future research

In this paper, the utility of dynamic pricing used to maximize the accumulated revenue of a firm was reviewed. In particular, a memoryless market with an infinite time horizon and infinite inventory scenario was considered. A new approach to measure the market price sensitivity better adapted to virtual market characteristics was proposed, and two sampling policies (Π_{ego} and Π_{dp1}) were adapted to work with non parametric inference models and compared to 3 other policies commonly found in literature. It has been shown that there is no significant difference between the proposed policies and the greedy policy. This motivates the authors to explore the design of new policies and the adaptation of known policies to the described paradigm. In particular, understanding the strengths of policies known to outperform the greedy policy in the traditional framework can provide insight on how to create better performing policies in the proposed setting. Future lines of research shall include the dynamic case where the market sensitivity changes with time –hinting to constantly consider exploration– and a delayed reward measurement case where the response of a sample is not known until after certain time.

References

1 Victor F. Araman and René Caldentey. Dynamic pricing for nonperishable products with demand learning. *Operations Research*, 57(5):1169–1188, 2009.

2 Victor F. Araman and René Caldentey. Revenue management with incomplete demand information. *Forthcoming in Encyclopedia of Operations Research*, 2011.

3 Yossi Aviv and Amit Pazgal. Pricing of short life-cycle products through active learning. *Working paper.*

4 Yossi Aviv and Amit Pazgal. A partially observed markov decision process for dynamic pricing. *Management Science*, 51(9):1400–1416, 2005.

5 Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., 1st edition, 2006.

6 Alexandre X. Carvalho and Martin L. Puterman. Learning and pricing in an internet environment with binomial demands. *Journal of Revenue and Pricing Management*, 3(4):320 – 336, 2005.

7 Guillermo Gallego and Garrett van Ryzin. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40(8):999–1020, 1994.

8 Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, December 1998.

9 P.K. Kannan and Praveen K. Kopalle. Dynamic pricing on the internet: Importance and implications for consumer behavior. *International Journal of Electronic Commerce*, 5(3):63 – 83, 2001.

10 Ken Littlewood. *Forecasting and control of passenger bookings.* American Airlines Inc, 1972.

11 David J. C. Mackay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, 1st edition, 2003.

12 David J.C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736, 1992.

13 Javad Nasiry and Ioana Popescu. Dynamic pricing with loss-averse consumers and peak-end anchoring. *Operations Research*, 59(6):1361–1368, 2011.

14 Carl E. Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.

15 Jaakko Riihimäki and Aki Vehtari. Gaussian processes with monotonicity information. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, AISTATS2010 special issue, 2010.

16 Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Pearson Education, 2nd edition, 2003.

17 Troy Wolverton. Amazon backs away from test prices, 2000. http://news.cnet.com/2100-1017-245631.html (Accessed 01/04/2012).

Empirical Bayes Methods for Discrete Event Simulation Performance Measure Estimation*

Shona Blair[1], Tim Bedford[1], and John Quigley[1]

1 **Dept. of Management Science, University of Strathclyde**
40 George Street, Glasgow, G1 1QE, UK
{shona.blair, tim.bedford, j.quigley}@strath.ac.uk

Abstract

Discrete event simulation (DES) is a widely-used operational research methodology facilitating the analysis of complex real-world systems. Although, generally speaking, simplicity is greatly desirable in DES modelling applications, in many cases the nature of the underlying system results in simulation models which are large in scale, complex, and expensive to run. As such, the careful design and analysis of simulation experiments is essential to ensure valid and efficient inference concerning DES model performance measures. It is envisaged that empirical Bayes (EB) methods, which enable data to be pooled across a set of populations to support inference of the parameters of a single population, may be of use within this context. Despite this potential, EB has so far been neglected within the DES literature. This paper presents a preliminary computational investigation into the efficacy of EB procedures in the estimation of DES performance measures. The results of this investigation, and their significance, are explored. Additionally, likely directions for future research are also addressed.

1998 ACM Subject Classification I.6.8 Types of Simulation

Keywords and phrases Discrete Event Simulation, Analysis Methodology, Empirical Bayes

Digital Object Identifier 10.4230/OASIcs.SCOR.2012.21

1 Introduction and Motivation

Discrete event simulation (DES) is a powerful and flexible methodology, widely utilized in OR applications for the design, analysis and improvement of complex, dynamic and stochastic real-world systems. At its core, DES involves abstracting the fundamental structure of the system of interest and using this information to construct a computer model of the system. A process of experimentation is conducted with the computer model in order to gain insight into and understanding of the performance of the real system. One of the key advantages of discrete event simulation is its ability to incorporate a "realistic" level of system complexity into the analysis process, when compared with the more rigid assumptions of alternative modeling techniques; indeed, DES is frequently referred to as a "method of last resort" [9]. Whilst the benefits of simple models are well understood and widely disseminated (see, for example, [11, 13]), there are many instances when the nature of the underlying system being studied necessitates the use of simulation models which are large-scale, structurally complex, difficult to interpret and computationally expensive to run. As such, the careful design and analysis of simulation experiments is necessary to ensure valid and efficient inference concerning model performance [8].

* This work was partially supported by an EPSRC CASE Studentship and SIMUL8 Corporation.

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn; pp. 21–30
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Empirical Bayes (EB) procedures offer a structured and theoretically sound framework for the pooling of data obtained across a set of populations to support inference concerning the parameters of an individual population. This often enables more efficient inference in situations which feature a repeated structure, providing that sufficient "similarity" exists between component populations. (For a general EB reference, see [2]). It seems intuitively reasonable that such an approach may be of benefit in simulation model experimentation, owing to the underlying similarity between simulation model configurations. In light of the computational expense involved in executing simulation models (touched on above), such increased efficiency in estimation would likely prove highly advantageous in practice. Yet, in spite of this apparent potential, EB has so far been neglected within the simulation literature.

This paper presents the results of a preliminary computational investigation into the use of EB procedures in the estimation of DES model performance measures. It begins, in Section 2, with the presentation and brief derivation of the EB procedures which are to be applied. Then, in Section 3, the DES model selected for testing is introduced, the reasons behind this choice outlined and certain theoretical results regarding performance measures of interest are presented. After this, in Section 4, the experimental design of the study is described, before the results are summarised in Section 5. The paper concludes, in Section 6, with some discussion of how this research area might be further explored in the future.

2 Introducing the Empirical Bayes Procedures

Empirical Bayes procedures feature a hierarchical model structure, identical to that of a traditional Bayesian analysis. As such, we typically have a situation in which model parameters are themselves represented by probability distributions, termed "prior" distributions. In a Bayesian analysis, the prior distribution would be subjectively determined, usually elicited from subject matter experts. However, in an empirical Bayes setting the data themselves determine the prior distribution. As mentioned in the previous section, EB methods are well-suited to applications featuring a large number of "similar" populations or processes. In this situation, the data obtained from each of the populations are pooled and used to provide inference on a general prior distribution. This general prior distribution is then combined with the individual samples from each of the populations using standard Bayesian updating to obtain a "posterior" distribution specific to each of the populations. (For a detailed overview of the above theory and terminology, please refer to [2].)

EB methods have a long history, with their roots in actuarial work on credibility theory, the first major publication by Robbins [12] in 1955 and a series of landmark papers in the 1970's by Efron and Morris [4, 5, 6] (see [10] for a more detailed account of their development). However, recent years have seen a huge upsurge in the volume of EB publications. This is due predominantly to scientific advances such as microarray technology, facilitating high-throughput biological screening and generating massive datasets that demand a fresh approach to statistical analysis [3]. A frequent feature of such datasets is their large number of populations, contrasted with relatively few observations from each. Such structures, as mentioned before, are ideally suited to an empirical Bayes analysis. Indeed, many successful applications have been published; a recent survey being [1]. Not surprisingly, this renewed interest has led to methodological and theoretical developments, as well as applications.

Here, however, we focus on some specific results which show particular promise in terms of their potential applicability within the context of DES model analysis. The empirical Bayes estimator shall employ is the “double-shrinkage” estimator presented in article [15]. This estimator assumes a normal/lognormal model of the data and its derivation is presented in the following subsections.

2.1 Assumptions

For $i = 1, 2, \ldots, p$, we assume:

$$X_i|\theta_i, \sigma_i^2 \overset{ind}{\sim} N(\theta_i, \sigma_i^2) \quad (1)$$

$$\theta_i \overset{iid}{\sim} N(\mu, \tau^2) \quad (2)$$

$$\log \sigma_i^2 \overset{iid}{\sim} N(\mu_v, \tau_v^2) \quad (3)$$

$$\log(S_i^2/\sigma_i^2) \overset{iid}{\sim} N(m, \sigma_{ch}^2) \quad (4)$$

where $m = E[\log(\chi_d^2)] = \psi(\frac{d}{2}) - \log(\frac{d}{2})$ and $\sigma_{ch}^2 = Var[\log(\chi_d^2)] = \psi'(\frac{d}{2})$, with d denoting the degrees of freedom. Initially, we suppose that hyperparameters μ, τ^2, μ_v and τ_v^2 are known.

2.2 Derivation

Given a sample of n observations, x_i, from sampling distribution (1) for population i, the prior distribution (2) on θ_i, and, for the moment, the additional assumption that σ_i^2 is known, a standard application of Bayes rule yields:

$$\theta_i|x_i, \sigma_i^2 \sim N(M_i\bar{x}_i + (1 - M_i)\mu, M_i\sigma_i^2), \quad (5)$$

where $M_i = \tau^2/(\tau^2 + \sigma_i^2/n)$ and $\bar{x}_i = \frac{1}{n}\sum_j x_{ij}$, for the posterior distribution of θ_i.

In such a case, we would use the posterior mean:

$$\hat{\theta}_i = M_i\bar{x}_i + (1 - M_i)\mu, \quad (6)$$

as a point estimator of θ_i.

However, the true population variances σ_i^2 for $i = 1, 2, \ldots, p$ are unknown, and as with [15] we adopt a lognormal prior (3) for σ_i^2, with the additional assumption (4) that S_i^2/σ_i^2 is also lognormally distributed (with parameters selected to coincide with those of χ_d^2/d, the standard distributional assumption regarding the quantity S_i^2/σ_i^2).

From (4), it follows that:

$$\log S_i^2 | \log \sigma_i^2 \sim N(m + \log \sigma_i^2, \sigma_{ch}^2), \quad (7)$$

and combining (3) and (7) using Bayes rule yields:

$$\log \sigma_i^2 | \log S_i^2 \sim N(M_v(\log S_i^2 - m) + (1 - M_v)\mu_v, M_v\sigma_{ch}^2), \quad (8)$$

where $M_\upsilon = \tau_\upsilon^2/(\tau_\upsilon^2 + \sigma_{ch}^2)$.

As in [15], we estimate this quantity using:

$$\hat{\sigma}_i^2 = \exp(M_\upsilon(\log S_i^2 - m) + (1 - M_\upsilon)\mu_\upsilon). \tag{9}$$

Thus, assuming known hyperparameters, we have the estimator:

$$\hat{\theta}_i = M_i\bar{x} + (1 - M_i)\mu, \tag{10}$$

with $M_i = \tau^2/(\tau^2 + \hat{\sigma}_i^2/n)$, where $\hat{\sigma}_i^2$ is as given by equation (9) above.

2.3 Estimation of Hyperparameters

All that remains is the estimation of the hyperparameters $\mu, \tau^2, \mu_\upsilon$ and τ_υ^2. As with [15], we adopt the following estimators.

For μ_υ and τ_υ^2, we have:

$$\hat{\mu}_\upsilon = \frac{1}{p}\sum_i(\log S_i^2 - m) \quad \text{and} \quad \hat{\tau}_\upsilon^2 = (\frac{1}{p}\sum_i(\log S_i^2 - m)^2 - \sigma_{ch}^2 - \hat{\mu}_\upsilon^2)_+,$$

from which we obtain:

$$\hat{M}_\upsilon = \frac{\hat{\tau}_\upsilon^2}{\hat{\tau}_\upsilon^2 + \sigma_{ch}^2} \quad \text{and} \quad \hat{\sigma}_{EB,i} = \exp(\hat{M}_\upsilon(\log S_i^2 - m) + (1 - \hat{M}_\upsilon).$$

To estimate μ and τ^2, we use:

$$\hat{\mu} = \frac{\sum_i(\bar{x}_i/\hat{\sigma}_{EB,i})}{\sum_i(1/\hat{\sigma}_{EB,i})} \quad \text{and} \quad \hat{\tau}^2 = (\frac{\sum_i(\bar{x}_i - \hat{\mu})^2}{p})_+.$$

Thus, the ‘double-shrinkage’ empirical Bayes point estimator is given as:

$$\hat{\theta}_{EB,i} = \hat{M}_{EB,i}\bar{x}_i + (1 - \hat{M}_{EB,i})\hat{\mu}, \tag{11}$$

where $M_{EB,i} = \hat{\tau}^2/(\hat{\tau}^2 + \hat{\sigma}_{EB,i}^2/n)$, and $\hat{\mu}$, $\hat{\tau}^2$ and $\hat{\sigma}_{EB,i}^2$ are as given above.

3 Introducing the DES Test Model

The purpose of this study is to evaluate the application of the EB methodology to the estimation of DES model performance measures. Having already introduced the EB procedures which are to be evaluated, this section aims to discuss an appropriate DES model upon which to test the EB procedures.

The model to be used is a computer-based implementation of an $M/M/1$ queuing model. This simple model consists of a single-server queuing system with exponentially distributed inter-arrival and service times and a first in - first out (FIFO) queuing discipline.

Simple "artificial" DES models such as this are frequently used in research for the evaluation of DES model analysis techniques [7]. These test models offer the key advantage that theoretical values are available for many performance measures of interest, and this knowledge greatly facilitates the testing of output analysis methods. One criticism which can be leveled at this approach is that such models bear little resemblance to the majority of DES models encountered in practice (those being significantly more complex). In light of this point, it is helpful to highlight the exploratory nature of the study; it should be emphasized that much of the value of this investigation lies in the issues it raises and the directions for further research which surface. This discussion is taken up again in Section 6 after the results are presented.

For now, some relevant queuing model theory and results, extracted from [14] and used in later sections of the paper, are presented.

3.1 $M/M/1$ Theory

As discussed above, an appropriate choice of DES test model for this investigation appears to be an $M/M/1$ model with a first in - first out queuing discipline. In this case, there are only two additional model parameters which may be varied, the arrival rate, denoted by λ, and the service rate, denoted by μ. In order to simplify our analysis, we note that these parameters may be combined to give a single parameter, namely the traffic intensity, which uniquely specifies a particular $M/M/1$ configuration. The traffic intensity parameter, denoted by ρ, is given by $\rho = \frac{\lambda}{\mu}$. We note that in future discussions, the particular $M/M/1$ configuration will be specified solely by reference to the traffic intensity, ρ.

Our performance measure of interest will be the steady-state (or long-run) expected time in system, which we denote by W. The exact value of this quantity for any $M/M/1$ model is given as a simple function of the arrival (λ) and service rate (μ) parameters, $W = \frac{1}{\mu-\lambda}$. This enables us to calculate the steady-state expected time in system exactly for any given $M/M/1$ model configuration.

4 Experimental Design

The aim of this section is to provide an overview of the experimental design employed during the computational testing. Although the key points will be presented, a more comprehensive picture of the research design may be obtained from visiting the following web address: `http://personal.strath.ac.uk/shona.blair/research/SCOR2012/`

Initially, it is important to mention that in this experiment, the performance of the EB estimator, $\hat{\theta}_{EB,i}$, (as given by (11) in Section 2), is evaluated relative to that of the standard estimator of a population mean, the sample mean $\bar{X}_i$. The sample mean is most prevalent point estimator of DES performance measures [9]; it offers the advantage of being easily interpretable and constitutes a convenient standard by which other methods may be compared.

In order to estimate the steady-state time in system, the test model had to be executed and time in system data collected. The relevant details concerning this stage of the experimentation are as follows:

- **Traffic intensities**: a mesh with lower limit $\rho = 0.02$, upper limit $\rho = 0.9$ and step-size $\rho = 0.02$ was used in the experiment. This resulted in 45 model configurations and gave a nice, fine grid.
- **Number of replications**: $200,000$ independent replications were made at each traffic intensity configuration to ensure sufficient data was collected.
- **Warm-up period**: a warm-up of 500 customers was used, ensuring that the model was in steady-state prior to data collection.
- **Run-length**: a run-length of 600 customers was used. The relatively short run-length (in comparison to the warm-up period) ensured the experiment reflected possible real-life data collection constraints.

This scheme yields a $200,000$ by 45 data matrix, where each data value represents an average time in system based on the first 100 customers after the aforementioned warm-up period.

Having obtained the necessary $M/M/1$ time in system data, we now discuss how this data can be used to calculate both EB and standard estimates. In the standard setting, we simply collect a batch of data from each traffic intensity configuration and calculate the sample mean to make inference regarding the true population mean. The only decision to be made concerns the size of the batch of data. In the empirical Bayes setting, however, data obtained from other model configurations can be pooled to support inference of the population mean of a given configuration. Thus immediate decisions need to be made concerning the size of the batches to be used and the pooling strategy to be employed.

To gain as much insight as possible, a range of batch sizes were explored, from 3 to 20. This enabled us to understand how the batch size affected the performance of the estimation technique. Additionally, in terms of the pooling strategy employed, a simple two sample t-test was conducted systematically for each pair of traffic intensity configurations to test for homogeneity. This approach was adopted as it avoids subjective decisions, based on theoretical knowledge of the system, biasing the results of the study. A range of significance levels, from $0.05, 0.1, 0.15, ..., 0.95$, were used to explore the relationship between pooling strategy and EB performance. This design resulted in 342 possible combinations of batch size and significance level, each of which was evaluated in the course of the computational testing. For each of these batch size / significance level combinations, the large volume of $M/M/1$ data collected permitted the calculation of $10,000$ pairs (both EB and standard) of estimates for each of the 45 traffic intensities. It is thus convenient to consider two $10,000$ by 45 matrices (one for each estimator) associated with each of the 342 batch size / significance level combinations.

To assess the performance of each estimator, the estimated values were subtracted from the true values, the errors squared, and the averages calculated over the 45 traffic intensities. This created, for each estimator, $10,000$ mean squared error (MSE) values. These were averaged, to remove the issue of stochastic variability, and the square root was taken to obtain the root mean squared error (rMSE) for each estimator. In order to gauge the relative efficacy, the ratio of the EB rMSE over the standard rMSE was taken as our critical statistical metric of interest. Finally, note that both the $M/M/1$ model and the EB and standard data analysis procedures were implemented in Matlab R2011a and run on a standard desktop PC (Intel Core-i5, 3GHz, 8GB RAM).

5 Summary and Discussion of Experimental Results

This section of the paper briefly presents and discusses the key results obtained from the aforementioned program of experimentation. As in the previous section, we begin by noting that a comprehensive set of numerical results and a detailed analysis may be found at the same web address. As mentioned in the previous section, the key statistical metric of interest is the ratio of the rMSE of the EB estimator to the rMSE of the standard estimator. The value of this quantity for each combination of batch size and significance level can found in Table 1 of Appendix A, which has been illustrated in Figure 1.

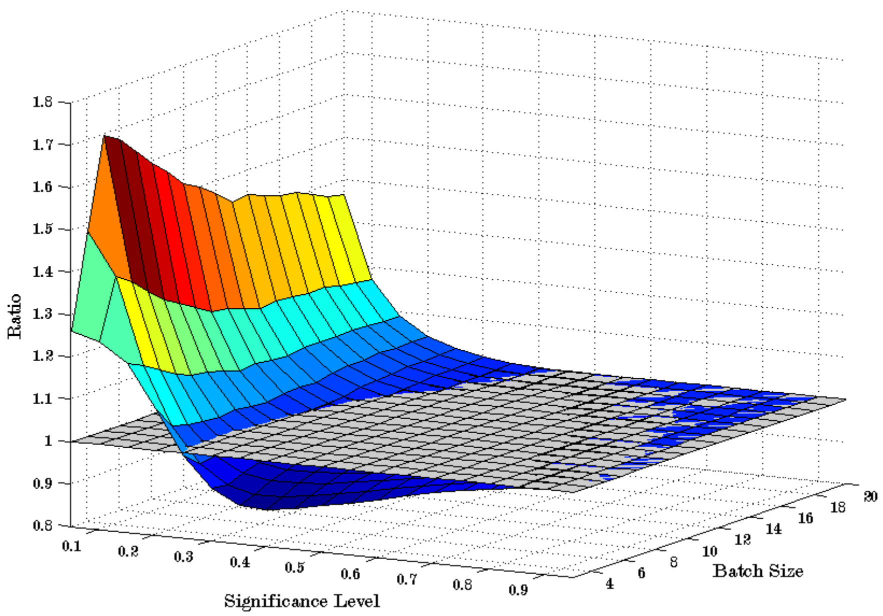


Figure 1 Ratio of EB rMSE to standard rMSE: This depicts the relative performance of the estimators, and how this varies with batch size and significance level. The plane at ratio= 1 enables us to easily identify areas where the EB estimator outperforms the standard estimator.

From examination of Table 1, the most favourable value of this ratio (0.8852) was obtained using a batch size of 3 and a significance level of 0.4, whilst the least favourable (1.6980) occurred for batch size 5 and a significance level of 0.05. Additionally, it appears that the optimal significance level, in terms of EB performance, is 0.4. As such, Figure 2 illustrates the relative EB performance for this significance level over varying batch size.

6 Conclusion and Future Research

As may be seen from the results illustrated in the previous section, quantifiable benefit can be achieved from the application of EB procedures to DES performance measure estimation.

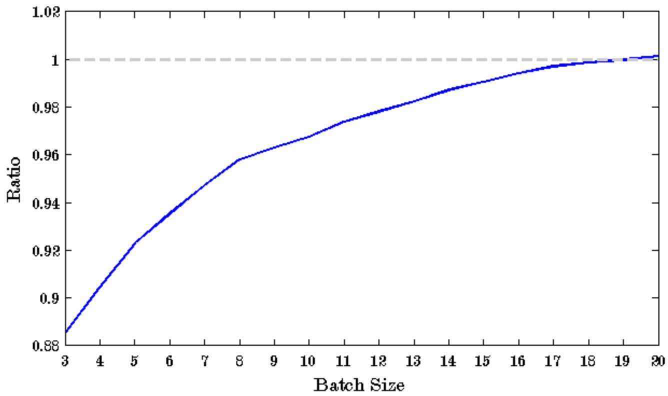


Figure 2 rMSE ratio over varying batch size for optimal significance level.

This is a positive outcome to our pilot study which goes a significant way towards establishing the feasibility of the application of empirical Bayes to DES model performance measure estimation. However, it is also apparent that the batch size used and the pooling strategy adopted are critical to the realization of this benefit. In this study, the decision as to whether or not to pool model configurations was made on the outcome of a simple two sample t-test. Although there are many more options for statistical tests of homogeneity, little in the way of formal guidance exists specifically concerning empirical Bayes pooling strategies. More exploration should be done on this subject with the aim of providing statistical indicators to guide practitioners in how to apply this method.

DES models exhibit great variety, differing vastly in characteristics such as purpose, structure, scale, nature of input parameters and nature of output distributions. This complexity increases the challenges involved in attempting to apply empirical Bayes to model analysis and in attempting to provide detailed guidelines enabling practitioners to make use of this methodology. A comprehensive study evaluating the benefits of EB in relation to more "realistic" DES models forms part of our ongoing research.

Acknowledgements We would like to thank Stephen Chick for several fruitful discussions on this topic and for his involvement as external supervisor of S. Blair, and the Simul8 Corporation for their technical advice and general project support.

A Detailed Results

Table 1 rMSE ratios for 342 combinations: EB outperforming standard highlighted.

Batch	Significance Level									
Size	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
3	1.2616	1.2425	1.1975	1.0992	**0.9934**	**0.9226**	**0.8887**	**0.8852**	**0.8939**	**0.9085**
4	1.4835	1.3842	1.1791	1.0449	**0.9616**	**0.9177**	**0.9026**	**0.9044**	**0.9134**	**0.9242**
5	1.6980	1.3588	1.1560	1.0339	**0.9666**	**0.9334**	**0.9215**	**0.9227**	**0.9290**	**0.9382**
6	1.6748	1.3238	1.1351	1.0287	**0.9711**	**0.9447**	**0.9346**	**0.9352**	**0.9393**	**0.9481**
7	1.6339	1.2898	1.1187	1.0255	**0.9782**	**0.9555**	**0.9471**	**0.9472**	**0.9506**	**0.9575**
8	1.5932	1.2686	1.1127	1.0302	**0.9877**	**0.9670**	**0.9585**	**0.9580**	**0.9602**	**0.9657**
9	1.5593	1.2473	1.1020	1.0265	**0.9891**	**0.9715**	**0.9635**	**0.9629**	**0.9651**	**0.9695**
10	1.5176	1.2244	1.0913	1.0243	**0.9907**	**0.9751**	**0.9682**	**0.9674**	**0.9697**	**0.9742**
11	1.4973	1.2189	1.0944	1.0311	**0.9990**	**0.9839**	**0.9763**	**0.9739**	**0.9755**	**0.9790**
12	1.4691	1.2055	1.0893	1.0323	1.0019	**0.9867**	**0.9801**	**0.9780**	**0.9790**	**0.9824**
13	1.4381	1.1922	1.0863	1.0312	1.0032	**0.9897**	**0.9842**	**0.9825**	**0.9826**	**0.9849**
14	1.4411	1.1983	1.0938	1.0391	1.0126	**0.9970**	**0.9900**	**0.9872**	**0.9870**	**0.9883**
15	1.4262	1.1947	1.0946	1.0444	1.0166	1.0018	**0.9939**	**0.9906**	**0.9904**	**0.9924**
16	1.4135	1.1900	1.0945	1.0439	1.0179	1.0042	**0.9976**	**0.9942**	**0.9932**	**0.9936**
17	1.4074	1.1910	1.0972	1.0492	1.0239	1.0092	1.0015	**0.9971**	**0.9961**	**0.9961**
18	1.3900	1.1816	1.0937	1.0476	1.0231	1.0094	1.0009	**0.9986**	**0.9965**	**0.9971**
19	1.3735	1.1769	1.0912	1.0484	1.0249	1.0119	1.0043	**0.9997**	**0.9979**	**0.9985**
20	1.3641	1.1718	1.0913	1.0496	1.0269	1.0131	1.0057	1.0013	**0.9992**	**0.9996**

Batch	Significance Level								
Size	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95
3	**0.9216**	**0.9359**	**0.9509**	**0.9651**	**0.9774**	**0.9878**	**0.9954**	**0.9993**	1.0000
4	**0.9368**	**0.9499**	**0.9618**	**0.9735**	**0.9838**	**0.9921**	**0.9975**	**0.9999**	1.0000
5	**0.9490**	**0.9591**	**0.9698**	**0.9803**	**0.9886**	**0.9947**	**0.9989**	1.0000	1.0000
6	**0.9569**	**0.9669**	**0.9761**	**0.9849**	**0.9915**	**0.9966**	**0.9993**	1.0000	1.0000
7	**0.9657**	**0.9737**	**0.9812**	**0.9885**	**0.9945**	**0.9983**	**0.9997**	1.0000	1.0000
8	**0.9718**	**0.9788**	**0.9861**	**0.9922**	**0.9967**	**0.9990**	**0.9999**	1.0000	1.0000
9	**0.9754**	**0.9820**	**0.9886**	**0.9937**	**0.9974**	**0.9995**	**0.9999**	1.0000	1.0000
10	**0.9788**	**0.9845**	**0.9907**	**0.9953**	**0.9981**	**0.9994**	1.0000	1.0000	1.0000
11	**0.9831**	**0.9884**	**0.9925**	**0.9964**	**0.9990**	**0.9999**	1.0000	1.0000	1.0000
12	**0.9859**	**0.9906**	**0.9945**	**0.9975**	**0.9995**	**0.9999**	1.0000	1.0000	1.0000
13	**0.9878**	**0.9928**	**0.9961**	**0.9985**	**0.9997**	1.0000	1.0000	1.0000	1.0000
14	**0.9913**	**0.9946**	**0.9974**	**0.9991**	**0.9999**	**0.9999**	1.0000	1.0000	1.0000
15	**0.9939**	**0.9962**	**0.9983**	**0.9996**	**0.9999**	1.0001	1.0000	1.0000	1.0000
16	**0.9950**	**0.9972**	**0.9991**	**0.9999**	1.0000	1.0001	1.0000	1.0000	1.0000
17	**0.9976**	**0.9995**	1.0003	1.0002	1.0000	1.0000	1.0000	1.0000	1.0000
18	**0.9979**	**0.9991**	**0.9997**	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000
19	1.0001	1.0006	1.0002	1.0000	1.0001	1.0000	1.0000	1.0000	1.0000
20	**0.9996**	1.0002	1.0002	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000

References

1 Y. Benjamini. Discovering the false discovery rate. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):405–416, 2010.

2 B.P. Carlin and T.A. Louis. *Bayesian methods for data analysis*. Texts in statistical science. CRC Press, 2009.

3 B. Efron. *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Institute of Mathematical Statistics Monographs. Cambridge University Press, 2010.

4 B. Efron and C. Morris. Limiting the risk of bayes and empirical bayes estimators - part ii: The empiricial bayes case. *Journal of the American Statistical Society*, 67:130–139, 1972.

5 B. Efron and C. Morris. Stein's estimation rule and it competitors - an empiricial bayes approach. *Journal of the American Statistical Society*, 68:117–130, 1973.

6 B. Efron and C. Morris. Dta analysis using stein's estimator and its generalizations. *Journal of the American Statistical Society*, 70:311–319, 1975.

7 K. Hoad, S. Robinson, and R. Davies. Classification of discrete event simulation models and output: creating a sufficient model set. In *Operational Research Society Simulation Workshop 2010 (SW10)*, 2010.

8 J.P.C. Kleijnen, S.M. Sanchez, T.W. Lucas, and T.M. Cioppa. State-of-the-art review: A user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing*, 17(3):263–289, 2005.

9 A.M. Law. *Simulation modeling and analysis*. McGraw-Hill series in industrial engineering and management science. McGraw-Hill, 2007.

10 J.S. Maritz. *Empirical Bayes methods*. Methuen's monographs on applied probability and statistics. Methuen, 1970.

11 M. Pidd. *Computer simulation in management science*. Wiley, 2004.

12 H. Robbins. An empirical bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, pages 157–164. University of California Press, 1955.

13 S. Robinson. Conceptual modelling for simulation part i: definition and requirements. *Journal of the Operational Research Society*, 59(3):278–290, 2008.

14 S.M. Ross. *Introduction to Probability Models*. Academic Press, 2009.

15 Z. Zhao. Double shrinkage empirical bayesian estimation for unknown and unequal variances. *Statistics and Its Interface*, 3(4):533–541, 2010.

The Transition to an Energy Sufficient Economy

Timothy S. Mbasuen[1] and Richard C. Darton[2]

1 **Department of Engineering Science, University of Oxford**
OX1 3PJ, Oxford, United Kingdom
`timothy.mbasuen@eng.ox.ac.uk`

2 **Department of Engineering Science, University of Oxford**
OX1 3PJ, Oxford, United Kingdom
`richard.darton@eng.ox.ac.uk`

Abstract

Nigeria is an energy-rich nation with a huge energy resource base. The country is the largest reserves holder and largest producer of oil and gas in the African continent. Despite this, only about 40% of its 158 million people have access to modern energy services. Around 80% of its rural population depend on traditional biomass. This paper presents an overview of ongoing research to examine energy policies in Nigeria. The aims are: 1) to identify and quantify the barriers to sustainable energy development and 2) to provide an integrated tool to aid energy policy evaluation and planning. System dynamics modelling is shown to be a useful tool to map the interrelations between critical energy variables with other key sectors of the economy, and for understanding the energy use dynamics (impact on society and the environment). It is found that the critical factors are burgeoning population, lack of capacity utilisation, and inadequate energy investments. Others are lack of suitably trained manpower, weak institutional frameworks, and inconsistencies in energy policies. These remain the key barriers hampering Nigeria's smooth transition from energy poverty to an energy sufficient economy.

1998 ACM Subject Classification I.6 Simulation and Modelling

Keywords and phrases energy sufficiency, energy policy, policy evaluation, economic planning, modelling

Digital Object Identifier 10.4230/OASIcs.SCOR.2012.31

1 Introduction

Nigeria, though an 'energy-rich' country falls within the category of countries suffering from 'energy poverty' as defined by the UN [1]. The country is the largest reserves holder and largest producer of oil and gas in Africa; yet, only about 40% of its total population have access to modern energy services [2]. Around 80% of its rural population depend almost wholly on traditional biomass for their energy needs [2]. The country's energy industry remains inefficient in meeting the energy needs and aspirations of its customers. This is evident in the dismal performance of the industry in terms of service delivery and per capita output.

For instance, in spite of its enormous oil and gas activities, the petroleum industry currently provides less than 15% of the country's GDP [3, 4]. Similarly, its GDP per capita and electricity consumption per capita remains among the lowest even in Sub-Saharan Africa-behind Angola and Ghana [5]. Persistent energy crisis has significantly weakened the industrialisation of the country, and grossly undermines the sustainable development agenda of the government.

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn; pp. 31–42
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A review of technical and policy documents [5, 6, 7, 8, 9], and discussions with key policymakers in the energy industry reveal that the most serious challenges hindering the country from meeting its energy aspirations include: burgeoning population, lack of capacity utilisation (at all levels), and inadequate energy investments. Others are corruption, lack of suitably trained manpower, weak institutional frameworks, and inconsistencies in energy policies [6]. These remain the key barriers hampering Nigeria's smooth transition from energy poverty to an energy sufficient economy.

Despite these enormous challenges, there is still the lack of appropriate policy evaluation and planning tools to aid informed decision making process. The only recognised structured tools that have so far been attempted in planning and examining energy policies in the country are the IAEA's[1] MAED[2] and WASP[3][10]. While these models are capable of giving valuable insights into analysis of energy demand and supply in an economy, they are not able to account for other dynamics relating to society and the environment, since they are largely based on a static economic modelling approach.

Elsewhere in the world, particularly in the transition and other developing economies, system dynamics models have been used to support energy policy evaluation and economic planning. Notable among recent examples is the successful application of the T21 model framework of the Millennium Institute (MI) in China, Denmark, the Balkans, and Guyana [11]. In the African continent and the Caribbean, specific examples include Ghana, Mali, Malawi, Mozambique, and Jamaica [11]. The success stories for these countries also serve as a motivation for developing a similar model for Nigeria.

In recognition of the overall long-term energy security and sustainability implications of the current energy/economic crises, the government has embarked on a policy of vigorous reforms to improve the situation. This paper presents an overview of ongoing research being undertaken to examine energy policies in Nigeria using system dynamics modelling and multi-criteria analysis. The aims are: 1) to identify and quantify the barriers to sustainable energy development and 2) to provide an integrated and holistic tool to aid energy policy evaluation and economic planning for the country.

2 The Barriers to Sustainable Energy Development in Nigeria

As an emerging economy, Nigeria has faced a number of development challenges. In its energy industry for example, there is currently a massive gap between energy demand and supply. The inability of the nation to effectively develop its vast energy resources, to improve the economic and the social wellbeing of its people is attributed to a range of barriers. The most fundamental of these factors are considered in this discussion.

2.1 Burgeoning Population

Nigeria's population has continued to grow over the last five decades; the total population was last reported at 158.32 million people in 2010, up from 110.00 million in 1995 [12]. In spite of birth control mechanisms currently available in the country-leading to a steady decline in birth rates, it is still expected that the country's overall population will continue to grow into the future. The implication however, is that a continuous rise in population

[1] International Atomic Energy Agency.
[2] Model for Analysis of Energy Demand: (for evaluation of future energy demand).
[3] Wien Automatic System Planning: (for finding optimal expansion plans for power generating systems).

means energy demand is likely to increase, even faster. Such a rise in population will not only constitute a serious barrier to energy poverty reduction strategy of the government, but will equally impede the overall human development process of the nation.

2.2 Lack of Capacity Utilisation

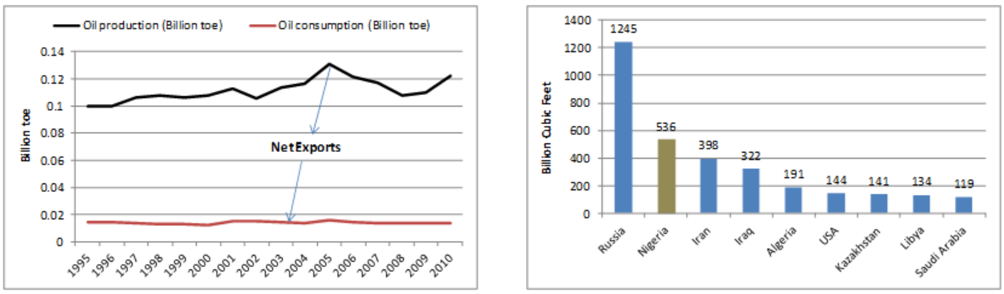
Despite its diverse energy resource base as illustrated in Table 1, the overall capacity utilisation of energy resources (defined in Btoe[4]) in the country remains very dismal. The key challenges include: lack of awareness, lack of skilled manpower, inadequate funding, and technological barriers [2, 13]. Lack of awareness in this context is taken to denote a low perception of the benefits of renewable and decentralised energy options. Furthermore, the inability of government to adopt state-of-the art energy efficient technologies such as CHCP[5] in its energy development strategies remains a serious impediment to sustainable energy development.

Table 1 Nigeria's Proved Conventional Energy Reserves at 2011 [7, 14].

SNo	Resource type	Reserve	Energy value (Btoe)	Lifespan (Years)
1	Crude oil	37.2 billion barrel	5.431	43.1
2	Natural gas	187 trillion scf	4.485	114
3	Coal and lignite	2.175 billion tonnes	1.512	79
4	Tar sands	40.6 billion barrel equiv.	5.928	50
5	Uranium	Not yet quantified	-	-

2.3 Inefficient Energy Supply Infrastructure

In spite of a relatively low energy consumption profile overall (Figure 1a), the in-country energy supply capability is still very depressing. For instance, the country maintains four refineries with a combined crude processing capacity of 450 000 barrels/day; yet, none of the refineries has ever operated to optimal capacity [14]. Over the last 15 years, some of the refineries operated at 0 and 30% capacity, leaving a shortfall of about 85% to be met by imports [14]. In the upstream sub sector, the country is known to substantially flare



(a) Oil Production and Consumption **(b)** Gas Flaring (2010)

Figure 1 Nigeria Oil Production & Consumption and Gas Flaring.

4 Billion tonnes of oil equivalent (authors' computations).
5 Combined Heat-Cooling and Power.

its natural gas for lack of infrastructure to market it. According to NOAA[6] [15], about 536 billion scf of natural gas was flared in the country in year 2010 (shown in Figure 1b). In monetary terms, the NNPC[7] estimates that gas flaring[8] costs the nation in average 2.5 billion US dollars per year in lost revenue [7]. While this is largely attributed to technological challenges, the problem is reinforced by inadequate funding, poor maintenance culture, and inconsistent energy policies.

2.4 Inadequate Energy Investments

Another barrier to effective development and utilisation of energy resources in Nigeria is inadequate funding. In the power sector for instance, government proposed to invest a total of 5.8 billion US dollars (for the period 2005 - 2013) [6]. This amount obviously, is inadequate, in view of the deplorable state of the electricity sub sector, and the high take-off investment funds required for developing renewable technologies.

The proposed public sector investment in the oil and gas upstream and downstream is approximately 3.8 billion US dollars [6]. In realistic terms, this amount is only sufficient to build one modern refinery. It implies therefore, that such meagre investment proposals can hardly make any noticeable difference, except more robust funding schemes are adopted.

2.5 Lack of Suitably Trained Manpower

The population structure of Nigeria provides a solid base for assessing availability and the quality of human capital and labour force necessary to transform the energy industry and the entire economy of the country. Review of energy industry and government reports [5, 6, 7], as well as historical data [14, 16] reveal that majority of the country's labour force are either unskilled or semiskilled. These categories of workforce apparently, are unlikely to change the economic fortunes of the country. Further discussions on this barrier are considered in the model representation section of this report.

2.6 Weak Institutional Frameworks

Many established energy institutions are unable to demonstrate focused regulatory and management capability required to improve energy development activities in the country. For example, the NNPC and DPR[9] have been proposing an end to gas flaring in the country for several years, but the deadlines to implement the policies has been repeatedly postponed [14]. Similarly, the loss of about 35 - 48% of electricity generated in the country to transmission and distribution losses is largely blamed on inability of the PHCN[10], to perform maintenance and upgrade of electricity supply infrastructure [2, 6]. These and many similar institutional challenges remain the barriers to smooth transformation of the nation's energy industry.

According to Transparency International's 2011 Corruption Perception Index (CPI), Nigeria is ranked 143 out of 182 countries for corruption-behind other African countries such as Botswana, Rwanda, and Namibia [17]. The evidence and impact of corruption can be seen in different sectors of the Nigerian economy. For example, 'Trading Economics' noted

---

[6] National Oceanic Atmospheric Administration.
[7] Nigerian National Petroleum Corporation.
[8] Gas Flaring in the context of oil production simply refers to the burning off (into the atmosphere) of natural gas associated with crude oil.
[9] Department of Petroleum Resources.
[10] Power Holding Company of Nigeria.

that Nigeria is one of the most advanced economies in Africa with average GDP growth of about 7.7% (as of last quarter of 2011), and about 95% foreign exchange earnings from oil and gas exports [18]. Yet, agriculture remains the main source of revenue to 2/3 of its population; while more than 50% of Nigerians still live in extreme poverty [12, 18]. Such dismal performance among other crucial issues is largely ascribed to corruption in both public and private sectors, and the lack of proper institutional frameworks for dealing with it [6, 14, 18].

However, the Economic and Financial Crimes Commission (EFCC) and other similar bodies in the country are working to address these issues [8, 9]. More detailed 'quantitative' discussions on corruption in the country is captured elsewhere in the model under 'Government Revenue and Expenditure'.

3 Energy Policy Reforms

As revealed in the foregoing, the current picture of energy development in Nigeria is marred by a number of challenges. It is against this backdrop that government has embarked on ambitious policy plans to improve the nation's energy and economic situation. One of the desired outcomes is to provide secure and sustainable energy access to its teeming population-particularly the rural population. To achieve this, the policy thrust indicates enhanced development of both conventional and renewable energy resources.

4 Current Study Objective and Methodology

4.1 Objective

The focus of this study is to develop a system dynamics (SD) model that maps the interrelations between critical energy variables (in Nigeria) and its impact on the three sustainability domains: the society, the economy, and the environment. The aims are: 1) to combine the SD model with a multi-criteria analysis (MCA) technique to examine energy policies, with the overall intent of identifying and quantifying the barriers to sustainable energy development. 2) To provide an integrated and holistic policy evaluation and economic planning tool to aid informed decision making in the country.

4.2 Methodology

This study consists of four major phases as follows:

Phase I: the first phase of the study (October 2010 – June 2011) involved critical review of literature on sustainability concepts; assessment tools and methodologies; current approaches to policy development and policy evaluation; and critique of the Nigerian energy system. The system dynamics methodology was found to be a useful approach for building an economy-wide dynamic model, which when combined with a simple, stakeholder-focused MCA method such as SMART[11], would provide a robust framework for analysing both the technical and social dynamics of the energy system.

Phase II: the second phase of the research (July 2011 – December 2011) developed a theoretical framework that enables integration of the proposed SD model with the chosen MCA technique. The idea was that combining an SD simulation model with a 'social-oriented'

[11] Simple Multi-Attribute Ranking Technique.

MCA model would result into a more rigorous and inclusive approach to exploring energy policies. The role of the MCA method is to account for stakeholder perspectives and to deal with issues of trade-offs. More importantly, it will aid us in making choices between competing policy alternatives.

Phase III: the third phase of the study (January 2012 – September 2012) is focused on data gathering, and building the proposed SD simulation model. The overall model conceptualisation and formulation process had commenced during the second phase of the research, with a few targeted Nigerian stakeholder contacts/discussions.

Phase IV: the fourth and last phase of the research (October 2012 – September 2013), will focus on full model verification, and a demonstration of how the SD simulation model and SMART value model can work together, for effective policy valuation. The process will initially involve systematic model calibration with historical data. In addition, the integrated model will be reviewed with experts in the ministries of energy and national planning (the potential users of the model), who earlier provided inputs in developing the model.

The verification process is also expected to be iterative, as it will largely involve presenting the model output to stakeholders, getting feedback, updating and representing it for additional comments. The idea is to ensure the most comprehensive input possible from experts and policymakers.

5 The Energy Dynamics Model

5.1 Model Purpose

In broad terms, the purpose of the proposed 'Energy Dynamics Model' is to map the interrelations that characterise the overall energy situation of Nigeria, and to serve as a 'baseline' from where alternative policies aimed at improving the social, the economic, and the environmental conditions of the country can be tested. In a specific sense, the model seeks to answer the overarching research question: *how and over what timescale can Nigeria make a smooth transition from 'energy poverty' to an 'energy sufficient' economy?*

5.2 Dynamical Hypothesis: a High Level Map

A high level map-HLM (Figure 2), was configured to encapsulate the overall model structure considered for the study. The HLM also serves as a dynamical hypothesis, which governs the development of the final model (using vensim™). In the HLM, there are four critical foci: energy resource (availability), energy demand (management), energy production/supply (efficiency), and energy financing (investment). The HLM hypothesises that these critical energy variables need to be broadly in balance, if Nigeria's transition to an energy sufficient economy is to be achieved smoothly. The elements are in a dynamic equilibrium, such that any distortion in their balance is likely to create ripple effects on the entire economy.

As illustrated in the map, availability (centre upper left corner) of viable primary energy resources (top left corner) is the first step towards ensuring energy security for the nation. The supply (centre lower right corner) of these resources is triggered by the demand (centre upper right corner) for energy to power and sustain the economy (top right corner). However, a sustainable production[12] (centre lower left corner) of the energy resources is only feasible

[12] Energy 'production' in this sense is distinguished from 'supply' in that by production, we refer to actual process of extracting/converting primary energy resources; while supply refers to the process of making energy products available (including domestic/exports).

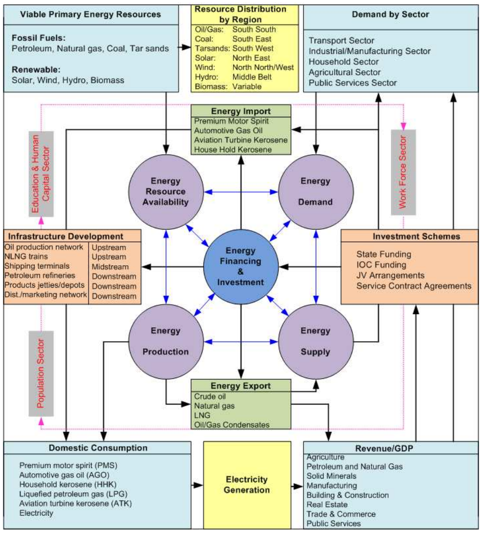


Figure 2 High Level Map: a dynamical hypothesis.

when enabled by continuous investments in energy development (captured in the nucleus). Other auxiliary sectors such as population, education, and workforce are also integrated to serve as drivers to the process.

The approach is not intended to be a definitive energy policy model, but the key significance remains that for Nigeria to break the bounds of energy poverty and put its energy industry on the path to sustainability[13], it is essential for the country to desire for such a balance.

5.3 Model Boundaries/Current Status

The model boundaries reflect energy impact on the three sustainability domains; it so far consists of six views. The first three views depict the core *social* sector variables, while the other views portray the three key *economic* sector variables interacting with the energy sector. The three views focusing on *environmental* impacts are currently being configured.

The model sectors as currently completed are: 1) population sector, 2) education and human capital sector, and 3) labour force sector. Others are 4) revenue/GDP sector, 5) government expenditure sector, and 6) technology sector. In this paper however, we present preliminary analysis on only two of the model views: education and human capital and the labour force sectors.

6 Model Representation

The overall model characterising the energy dynamics is developed for a 30 year timescale (1995–2025). Two simplified views of the model as mentioned above are presented below. The data that provides input parameters for the model are collected in EXCEL spreadsheets. They are however, inputted into the model as lookups, or as imported datasets.

[13] Energy/Economic sustainability in this context refers to dependable energy supply through efficient production and consumption in a thriving economy, with minimum environmental impact.

6.1 Education and Human Capital Sector

This model view focuses on the need for adequate supply of suitably qualified people through education and training.

As shown in the stock and flow diagram (Figure 3), the education sub-model starts with inflow of people (from the corresponding population cohorts) into primary education. A portion of the people who successfully complete primary school (after 6 years), progress to secondary education, while the remaining fraction (of primary school leavers) become available as *pool of labour force with primary education.* This trend continues until a fraction of graduates attain professional qualifications either in arts/commercial or scientific disciplines.

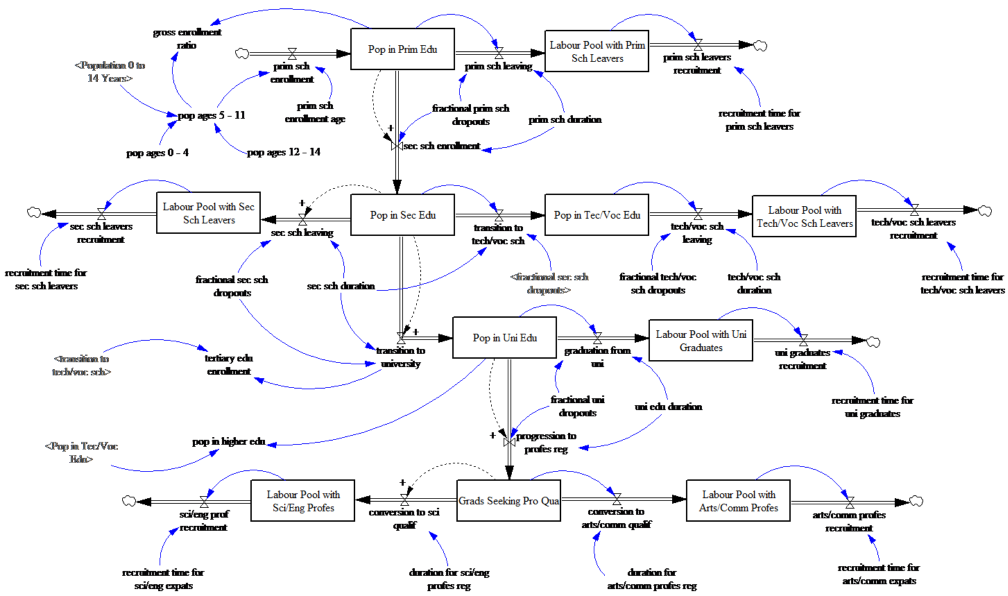


Figure 3 Education and Human Capital Sub-model.

6.2 Workforce Sector

The model view (Figure 4) describes the quality of workforce available in Nigeria. It defines four categories of labour force according to their highest level of education/skills. These include: 1) unskilled workforce (primary/secondary level); 2) semiskilled workforce (technical/vocational level); 3) skilled workforce (university level); and 4) highly skilled workforce (professional level).

7 Model Verification

A specific time is allotted in the research plan for full verification of the model structure and simulated behaviour through experts feedback in Nigeria, which in the context of this study is the most important 'validation'. However, for the purposes of preliminary analysis and in verifying the model's dimensional consistency, initial calibration of completed model views with historical time series data was undertaken. Fifteen years (1995 - 2010) data were inputted into the model as datasets, to systematically verify the model behaviour with what is known. The calibrations are however, not presented here for reason of space, but they generally show a high degree of congruency.

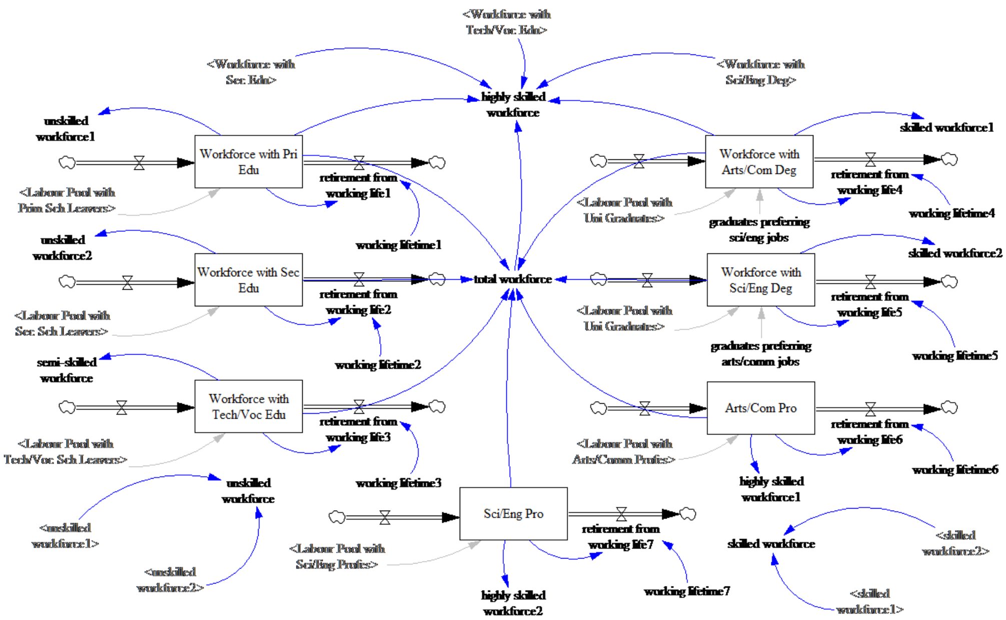


Figure 4 The Workforce Sub-model.

8 Preliminary Analysis

The outputs for model views considered in this report (in base runs) are discussed below.

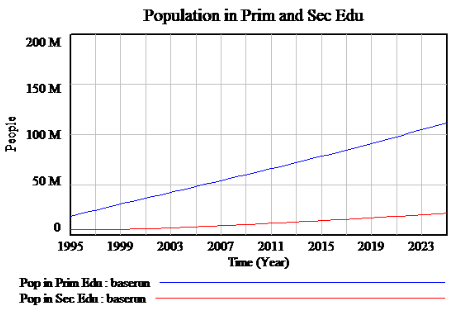
8.1 Simulation Run-1: Education and Human Capital

This model run features the supply of human capital based on educational attainment. The population of people in primary and secondary education for example, changed from 18.58 million and 5.121 million people in 1995 to 62.92 million and 10.90 million in 2010 respectively. As shown in Figure 5a, there is a wide gap between the population in primary education and those in secondary education. This implies that a significant number of people leaving primary education fail to progress to secondary school, perhaps, due to lack of awareness, motivation, or economic hardship. Also in Figure 5b, the population of people preferring to attend university education after secondary as opposed to technical/vocational studies is growing.

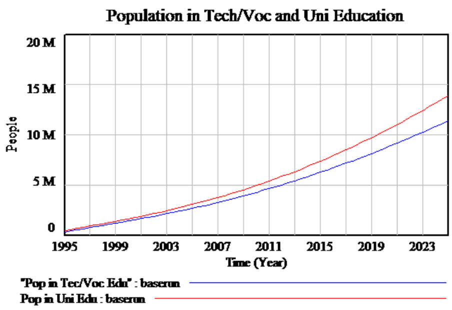
This trend became pronounced in the post 2000 scenarios because of recent 'discriminatory attitudes' of employers (particularly the private sector), in showing preference to university graduates. To that effect, the number of graduates seeking further professional qualifications is also rising (Figure 5c). However, a combination of overall educational attainments (Figure 5d), reveal that majority of the population terminate education after primary school. In view of Nigeria's population size, the trend is becoming worrisome as it is giving rise to unprecedented numbers of unskilled labour force in the country.

8.2 Simulation Run-2: Labour Force

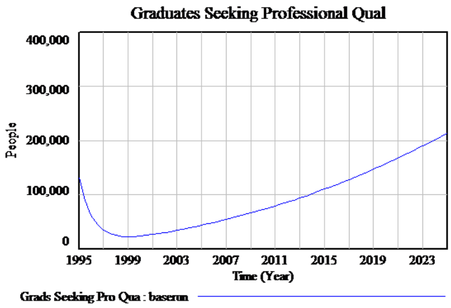
Despite a huge gap between the population progressing to further education and those terminating after primary school, it is glaring in Figure 6a, that the people qualifying to secondary education constitute the largest share of total workforce. Also, university graduates



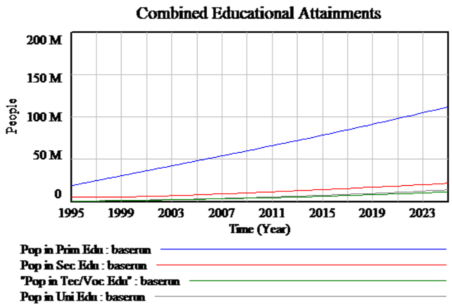
(a) Population in Prim & Sec Edu



(b) Population in Tech/Voc & Uni Edu

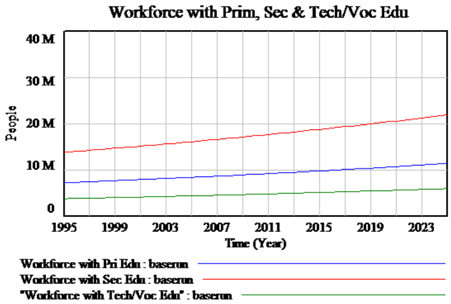


(c) Graduates Seeking Profsnl Qual

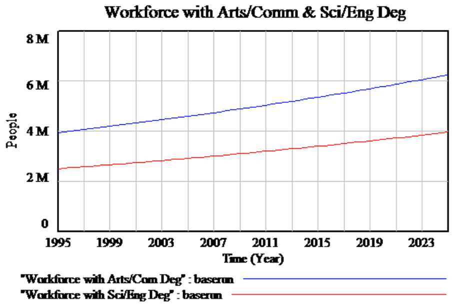


(d) Combined Edu Attainments

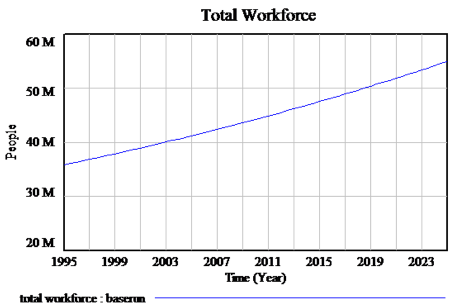
Figure 5 Model Output for Education and Human Capital.



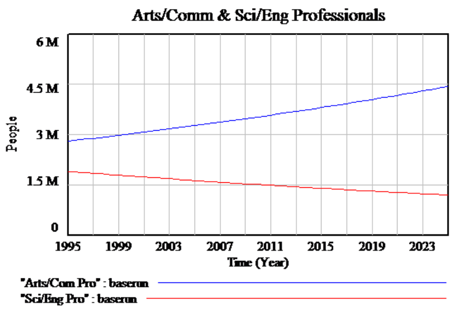
(a) Workforce with Prim/Sec/Tech/Voc



(b) Workforce with Arts/Comm & Sci/Eng



(c) Total Labour Force



(d) Skilled Arts/Comm & Sci/Eng Profsnls

Figure 6 Model Output for Workforce.

with arts/commercial degrees are more than those with science and engineering degrees (Figure 6b). In spite of an overall growth in its workforce (Figure 6c), it is still evident (Figure 6d), that the population of highly skilled professionals in arts/commercial disciplines is growing much faster, while that of scientific disciplines is somewhat on the decline.

In reality, the population with primary, secondary and technical/vocational educational qualifications currently dominate the Nigerian energy sector. They constitute more than 60% of the total labour force, as opposed to skilled and highly skilled labour force. The overall implication is that these semi- and un-skilled categories of workforce are not well trained, and this leads to the general lack of people with advanced technical and leadership skills who can drive development of the energy economy.

9 Summary Findings

Our model, though a work in progress, has provided insights to the barriers to sustainable energy and economic development of Nigeria. It raises concerns about the quality of human capital and workforce available to influence economic revolution in the country. The model emphasises the potential for human capital development, but underscores weak educational system and lack of suitable technical skills as ongoing concerns. Although, no policy scenarios have been generated at this stage, the model foresees the needs for enhanced education and capacity development at all levels.

10 Conclusions and Future Work

Engagements with system dynamics (as a policy evaluation and planning tool) are rare in Nigeria, as there is currently no account of its application anywhere in the country. For that reason, this study becomes novel for its introduction of an integrated approach to examining policy issues, and for providing a tool that will aid informed decision making process in the country. Our approach seeks to combine the quantitative rigour of system dynamics modelling and the social focus of the chosen MCA technique in planning and evaluation of energy/economic policies. To help in charting a new path to a sustainable future for a developing nation like Nigeria; it is essential that such a holistic and integrated tool be provided.

The study focus over the final phase of the research is to complete development of the model sectors, and have it run as an integrated unit. In its full running mode, the model will be used to generate and test different policy intervention scenarios based on selected Nigerian government targets. The emerging policy options (from the model) will further be evaluated (with MCA) against some criteria to determine their robustness.

References

1 United Nations Commission on Sustainable Development. *UNCSD report of the ninth session*. UNCSD., New York, USA, 2001

2 Sambo, A.S. *Matching Electricity Supply with Demand in Nigeria*. Sambo., available online at: www.energy.gov.ng

3 International Monetary Fund. *IMF Annual Report: Supporting a Balanced Global Recovery*. International Development., Washington DC, 2010

4 National Bureau of Statistics. *Statistical News: Trade Statistics*. NBS., Abuja, Nigeria, 2010

5 Central Bank of Nigeria. *Annual Reports and Statement of Accounts*. CBN., Abuja, Nigeria, 2009

6 Federal Ministry of National Planning. *Nigeria Vision 20:2020 First Implementation Plan (2010 - 2013)*. FMNP., Abuja, Nigeria, 2010

7 Nigerian National Petroleum Corporation. *Oil and Gas Statistics: Annual Statistical Bulletin*. NNPC., Abuja, Nigeria, 2011

8 Economic and Financial Crimes Commission. *Fighting Corruption in Nigeria*. EFCC., available online at: www.efccnigeria.org/

9 Independent Corrupt Practices and Other related Offences Commission. *Fighting Corruption in Nigeria*. ICPC., available online at: www.icpc.gov.ng/

10 Sambo, A. S., Iloeje, O. C., Ojosu, J. O., Olayande, J. S., Yusuf, A. O. *Nigeria's Experience on the Application of IAEA's Energy Models (MAED & WASP) for National Energy Planning*. Korea Atomic Energy Research Institute, Daejon, Korea, 2006

11 Millennium Institute. *T21 Models, Papers and Newsletters*. MI., available online at: www.millennium-institute.org/resources/elibrary/index.html

12 United Nations Development Programme. *Human Development Report*. UNDP., Washington DC, 2010

13 Global Network on Energy for Sustainable Development. *Renewable Energy Technologies and Poverty Alleviation: Overcoming Barriers and Unlocking Potentials*. GNESD., available online at: www.gnesd.org/publications.htm

14 US Energy Information Administration. *Country Analysis Briefs: Nigeria*. EIA Energy and Mining Report., Washington DC, 2011

15 National Oceanic and Atmospheric Administration. *Global Gas Flaring Country Data*. National Geographical Data Center., USA, 2010

16 IBRD/World Bank. *Africa Development Indicators 2010*. IBRD., Washington DC, 2010

17 Transparency International. *Countries Ranking: Corruption Perception Index 2011*. TI., available online at: www.transparency.org

18 Trading Economics. *Indicators: Nigeria GDP Growth Rate*. TE., available online at: www.tradingeconomics.com/nigeria/gdp-growth

Can online trading algorithms beat the market? An experimental evaluation

Javeria Iqbal[1], Iftikhar Ahmad[1], and Günter Schmidt[1,2]

1 Chair of Information and Technology Management
University of Saarland, Germany
2 Department of Statistical Sciences
University of Cape Town, South Africa
{ji,ia,gs}@itm.uni-sb.de

Abstract

From experimental evaluation, we reasonably infer that online trading algorithms can beat the market. We consider the scenario of trading in financial market and present an extensive experimental study to answer the question "Can online trading algorithms beat the market?". We evaluate the selected set of online trading algorithms on $DAX30$ and measure the performance against buy-and-hold strategy. In order to compute the experimentally achieved competitive ratio, we also compare the set of algorithms against an optimum offline algorithm. To add further dimensionality into experimental setup, we use trading periods of various lengths and apply a number of evaluation criteria (such as annualized geometric returns, average period returns and experimentally achieved competitive ratio) to measure the performance of algorithms in short vs. long term investment decisions. We highlight the best and worst performing algorithms and discuss the possible reasons for the performance behavior of algorithms.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases Online Algorithms, Experimental Evaluation, Competitive Analysis

Digital Object Identifier 10.4230/OASIcs.SCOR.2012.43

1 Introduction

The major concern of an investor (both individual and corporate) in financial market is the profitability of the underlying trading strategy. As financial markets are highly volatile and risky, an investor wants to ensure that the trading strategy that he/she relies on, is profitable and provides better returns in unforeseen circumstances. Performance evaluation of the trading strategies is required to ensure that the selected strategies are thoroughly tested in real world scenario. Trading in financial markets is based on the principle of maximizing the difference between selling and buying, i.e., buying at minimum possible price and selling at maximum possible price. However, as the investor (henceforth called as player) cannot see the future prices, his decision is based on limited knowledge.

Online algorithms can be used to support trading decisions in financial markets where complete knowledge about future is not available. Online algorithms are based on the paradigm that the player has no knowledge about future and every decision the algorithm makes is based on the current knowledge of the algorithm. Online algorithms are evaluated through competitive analysis paradigm. Competitive analysis is used to measure the performance of an online algorithm against an optimum offline algorithm. Let $\mathcal{P}$ be a maximization problem and $\mathcal{I}$ be the set of all input instances, ON be an online algorithm for problem $\mathcal{P}$

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn; pp. 43–52
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and $ON(I)$ be the performance of algorithm ON on input instance $I \in \mathcal{I}$. Let OPT be an optimum offline algorithm for the same problem $\mathcal{P}$, ON is said to be *c-competitive* if $\forall I \in \mathcal{I}$

$$\mathcal{ON}(I) \geq \frac{1}{c} \cdot \mathcal{OPT}(I). \tag{1}$$

We consider a set of online algorithms for trading and evaluate them from empirical perspective by executing the set of algorithms on real world. We evaluate the performance of algorithms using different lengths of trading periods (10, 20, 60, 130 and 260 days) and use a variety of evaluation measures such as annualized geometric returns (AGR), average period return (APR) and experimentally achieved competitive ratio (c_e) to evaluate the selected set of online algorithms.

Definitions: We present a set of basic terms and definitions with reference to online trading algorithms.

i. *Transaction*: A transaction is either selling *or* buying of an asset.
ii. *Trade*: A trade consists of two transactions, one is buying and one is selling. The number of trading periods in an investment horizon is represented by P.
iii. *Investment Horizon*: The total time duration in which all transactions are carried out. The investment horizon can be divided into one or more time segments for trading.
iv. *Duration (T)*: The length of a trading period.
v. *Offered Price (q_t)*: The offered price on day t.
vi. *Upper Bound (M)*: The upper bound on possible offered prices during the trading period.
vii. *Lower Bound (m)*: The lower bound on possible offered prices during the trading period.
viii. *Amount Converted (s_t)*: Specifies which fraction of the amount available (e.g. wealth) is to be converted at price q_t on day t, with $0 \leq s_t \leq 1$.

2 Literature Review

In this section, we briefly describe different performance evaluation techniques for online algorithms and provide a short review of the literature in relation to experimental analysis of different trading strategies.

The performance of online algorithms are analyzed from three different perspectives, *Bayesian Analysis* assumes that the input instance is drawn from a known probability distribution and expected performance of the algorithm is investigated on the assumed probability distribution, *Competitive Analysis* compares the performance of online algorithms against that of optimum offline algorithm, and *Experimental Analysis* uses the back testing technique to evaluate the performance of online algorithm on real world (as well as synthetic) data. Each of these techniques has its own limitations and drawbacks. The Bayesian analysis is considered to be too optimistic and relies heavily on underlying distribution. The competitive analysis is criticized to be too pessimistic as online algorithm is compared with an optimum offline algorithm which has complete knowledge about the future, thus experimental analysis can be used as a useful tool in conjunction with worst case competitive analysis to evaluate the performance of online algorithms and to measure the disparity between theoretical and real world performance of online algorithms.

The experimental analysis of online trading algorithms is not widely addressed in the literature. Mohr and Schmidt [7] presented an empirical study where the reservation price algorithm of El-Yaniv et al. [2] is compared with buy-and-hold (BH). Schmidt et al. [9] evaluated the reservation price algorithm and threat based algorithm [2] to dollar average strategy (DAS) and BH. However, there is no such study which considers a broader set of online algorithms for trading and evaluate them extensively on real world dataset.

The experimental evaluation of heuristics trading algorithms has received considerable attention from researchers. In these studies BH is used as a benchmark for comparing different strategies. Due to space constraint the details cannot be included and the reader is referred to [1, 4, 6, 8, 10, 11].

3 Online Trading Algorithms

We briefly discuss the set of online trading algorithms. The proof of competitive ratio and optimality of algorithm is left out due to space constraint and the reader is referred to corresponding research papers. The online trading algorithms are broadly classified as *Non Preemptive* and *Preemptive* algorithms.

3.1 Non Preemptive Algorithms

In non preemptive algorithms (also referred to as reservation price algorithms), the player invests (buys or sells) his whole wealth at one point of time in investment horizon. The player computes a reservation price and compares each offered price with reservation price. A reservation price is the maximum(minimum) price which a player will accept for buying (selling). The player takes a buy decision and invests at the offered price if it is less or equal than the pre-computed reservation price, and a sell decision is taken when the offered price is greater (or equal) than the pre-computed reservation price. We discuss algorithms presented by El-Yaniv et al. [2] and Kao and Tate [3] and BH strategy.

3.1.1 El-Yaniv et al. reservation price algorithm

El-Yaniv et al. [2] presented reservation price algorithm based on the assumption that the lower and upper bound of offered price m and M are known to the player. El-Yaniv et al. presented the strategy for max-search (sell) problem. Schmidt et al. [9] analogously extended the strategy for buy. The buy and sell strategies are given as following;

- **Algorithm 1** (*RPMm*)
  - Buy at the first price less than or equal to $q^* = \sqrt{M \cdot m}$.
  - Sell at the first price greater than or equal to $q^* = \sqrt{M \cdot m}$.

3.1.2 Kao and Tate online difference maximization approach

Kao and Tate [3] presented a non-preemptive strategy maximizing the difference between the rank of selected buy and sell items. Let x_t be the rank of q_t in the t prices observed so far, Kao and Tate [3] described the buy and sell strategies as following;

- **Algorithm 2** (*KT*)
  - Buy at price q_t if $x_t \le \mathcal{L}_T(t)$.
  - Sell at price q_t if $x_t \ge \mathcal{H}_T(t)$.

Where $\mathcal{L}_T(t)$ and $\mathcal{H}_T(t)$ are the limits (reservation price) for buy and sell respectively and are calculated as following;

$$\mathcal{H}_T(t) = \left\lceil \frac{t+1}{T+1} \cdot R_T(t+1) \right\rceil . \tag{2}$$

$R_T(t)$ is the expected final rank of high selection (sell) if the optimal strategy is followed starting at the t-th time. $R_T(t)$ is calculated as following;

$$R_T(t) = \frac{\mathcal{H}_T(t) - 1}{t}\left(R_T(t+1) - \frac{T+1}{2(t+1)}\mathcal{H}_T(t)\right) + \frac{T+1}{2}. \tag{3}$$

$$\mathcal{L}_T(t) = \begin{cases} 0 & t = T, \\ \left\lfloor \frac{t+1}{T+1} \cdot (R_T(t+1) - P_T(t+1)) \right\rfloor & t < T. \end{cases} \tag{4}$$

$P_T(t)$ is the expected high-low (sell-buy) difference, following optimal strategy at step t, $P_T(t)$ is calculated as following;

$$P_T(t) = \begin{cases} 0 & t = T, \\ P_T(t+1) + \frac{\mathcal{L}_T(t)}{t}\left(R_T(t+1) - P_T(t+1) - \frac{T+1}{t+1} \cdot \frac{\mathcal{L}_T(t)+1}{2}\right) & t < T. \end{cases} \tag{5}$$

3.1.3 Buy-and-Hold

The buy-and-hold strategy is a long term investment policy which is used as a benchmark in financial markets for comparing the performance of other trading algorithms and strategies. The key idea is that financial markets returns are worthwhile in contempt of volatility and recession.

- **Algorithm 3** (*BH*)
  - Buy at the the first offered price q_1.
  - Sell at the the last offered price q_T.

3.2 Preemptive Algorithms

In preemptive algorithms, the player does not invest (buy and sell) at one point of time in investment horizon, instead the player invests a portion of wealth. The exact investment amount depends on the offered price and/or time of investment. We consider the preemptive algorithms of El-Yaniv et al. [2], Lorenz et al. [5] and dollar average strategy.

3.2.1 El-Yaniv et al. threat based algorithm

El-Yaniv et al. [2] strategy is based on the assumption that the adversary may drop the offered price to some minimum level m and will keep it there for the rest of investment horizon, the strategy maximizes the performance of the algorithm while safeguarding itself against the assumed threat. The basic rules of the algorithm are as following;

- **Algorithm 4** (*YFKT*)
  1. Consider a conversion from asset D into asset Y only if the price offered is the highest (lowest for selling) seen so far.
  2. Whenever convert asset D into asset Y, convert just enough D to ensure that a competitive ratio c would be obtained if an adversary drops the price to the minimum possible price m, and keeps it there afterwards.
  3. On the last day T, all remaining D is converted into Y, possibly at price m.

El-Yaniv et al. [2] proposed different variants of Algorithm 3.2.1, each assuming different a-priori information. We consider the variant of Algorithm 3.2.1 where the player has knowledge about lower and upper bounds (m and M) of offered prices.

3.2.2 Lorenz et al. algorithm

Lorenz et al. [5] proposed a strategy where the player has the information about the lower and upper bounds of offered prices. Two different strategies were proposed, one each for buying and selling.

- **Algorithm 5** (*LPS*)
  - Selling (Max-search) Problem: At the start of the game the player computes reservation prices $q_i^* = (q_1^*, q_2^*, ...q_u^*)$, where $i = 1, .., u$. As the prices are observed by the player, he accepts the first price which is at least q_1^*. The player then waits for the next price which is at least q_2^*, and so on. If there is still some wealth left on day T, it must be sold at the last offered price, which may be at the lowest price m.

$$q_i^* = m\left[1 + (c-1)\left(1 + \frac{c}{u}\right)^{i-1}\right]. \tag{6}$$

    Where c is the competitive ratio for the max-search (selling) problem.
  - Buying (Min-search) Problem: Follows the same procedure as for max-search problem, the reservation prices are computed as follows;

$$q_i^* = M\left[1 - \left(1 - \frac{1}{c}\right)\left(1 + \frac{1}{u \cdot c}\right)^{i-1}\right]. \tag{7}$$

    Where c is the competitive ratio for the min-search (buying) problem.

3.2.3 Dollar average strategy (*DAS*)

The dollar average strategy is based on fixed investments on predefined time intervals. We apply DAS to invest equal amount of wealth on each day of the investment horizon. Let $L = T/2$ be the length of sell (buy) period, we invest $1/L$ of the total wealth on each day.

4 Experimental Evaluation

We discuss the basic experimental setup, dataset, assumptions and results as follows;

4.1 Settings

We discuss the datasets utilized for performance analysis, the assumptions considered for the trading algorithms and methodology adapted for evaluation process.

Dataset: We consider real world dataset of $DAX30$ (Jan 1st 2001 - Dec 31st 2010) for experimental evaluation.

Assumptions: We consider the following set of assumptions.

1. The initial wealth assumed is always 1 unit.
2. All prices are daily closing prices.
3. In Frankfurt stock exchange, the transaction cost is calculated as $\min\{\max\{0.60, 0.0048\%\}, 18\}$, i.e 0.0048% of the market value (minimum of 60 cents) upto maximum of 18 euros. Thus, the transaction costs are highly dependent on amount transacted, but for the sake of simplicity and other factors such as liquidity (which is limited in stock market) we consider a transaction cost of 0.0048% of volume traded.

4. The length of trading period is T $\in$ {10,20,60,130,260} days.
5. The interest rate considered is zero as wealth (money) deposited in broker account is interest free.
6. We assume that the algorithm knows the a-priori information such as m and M etc, as this is required for the proper working of algorithm.

Evaluation Criteria: We use annualized geometric returns (AGR), average trading period return (APR), experimentally achieved competitive ratio (c_e), and average number of transactions (Tx) as criteria to evaluate the performance of trading algorithms. The different measures provide a deeper insight on profitability of an algorithms in short and long runs. We discuss the number of transactions performed by each algorithm to discuss the potential impact of transaction cost on performance of algorithm.
Let D_i and d_i be the amount of wealth at the start and end of trading period i. Return of the trading period i, is given as;

$$r_i = d_i/D_i. \tag{8}$$

Geometric return is based on the assumption that the wealth at hand at start of the period i is invested in the next period $i+1$. Geometric returns can be used to evaluate the annualized performance that algorithm achieves in investment horizon. Let P be the number of trading periods in an investment horizon, if the number of years in the investment horizon, $y \geq 1$, then AGR is calculated as;

$$AGR(P) = \left(\prod_{i=1}^{P} r_i\right)^{1/y}. \tag{9}$$

The average period return (APR) is used for performance evaluation of algorithms, where we assume trading periods of same length and averages the results over all trading periods of same length. APR reflects the expected average performance within a trading period of given length.

$$APR(P) = \left(\prod_{i=1}^{P} r_i\right)^{1/P}. \tag{10}$$

Although, AGR and APR provides a useful insight about the profitability of an algorithm, it does not tell the whole story, as it is a stand alone measure and fails to measure the performance of an algorithm against the optimum possible result. Experimentally achieved competitive ratio (c_e) measures the performance of an algorithm against that of the optimal algorithm and can be use in conjunction with AGR and APR to report the profitability of an algorithm.

4.2 Results

In the following, we present the performance evaluation of algorithms on $DAX30$. We analyze the average number of transactions as well.

Performance evaluation: We discuss annualized geometric returns (AGR), average period return (APR) and experimentally achieved competitive ratio (c_e).
Table 1 describes the AGR of algorithms on $DAX30$ dataset for trading periods of different lengths. It can be observed that the performance of YFKT is the best whereas BH and

Table 1 Annualized Geometric Return (AGR) over $DAX30$.

Period	10	20	60	130	260
OPT	1.6976	1.7317	1.5165	1.4180	1.2694
RPMm	1.3715	1.2336	1.1493	1.0982	1.0851
KT	0.7698	0.8599	0.9822	1.0261	1.0172
BH	0.7485	0.8654	0.9634	0.9729	0.9852
YKFT	**1.4411**	**1.4112**	**1.2525**	**1.2091**	**1.1585**
LPS	0.9733	1.0526	1.0654	1.0909	1.0319
DAS	0.7592	0.8729	0.9637	1.0068	0.9870

Table 2 Average Period Return (APR) over $DAX30$.

Period	10	20	60	130	260
OPT	1.0206	1.0431	1.1097	1.1908	1.2694
RPMm	1.0122	1.0163	1.0354	1.0479	1.0851
KT	0.9900	0.9885	0.9955	1.0130	1.0172
BH	0.9889	0.9889	0.9907	0.9864	0.9852
YKFT	**1.0142**	**1.0268**	**1.0579**	**1.0996**	**1.1585**
LPS	0.9990	1.0040	1.0160	1.0445	1.0319
DAS	0.9895	0.9896	0.9908	1.0034	0.9870

DAS are least profitable strategies. The performance of RPMm is second only to YFKT. The performance of algorithms in term of "average period return" on DAX30 dataset is summarized in Table 2. The performance of threat based algorithm of El-Yaniv et al. [2] (YFKT) is the best among all online algorithms. YFKT performs consistently better over trading periods of different lengths. The BH is least productive algorithm with minimum returns for all trading periods except on trading period of length 20, where Kao and Tate [3] (KT) return is the minimum among the set of considered algorithms. Figure 1 summarizes the performance of algorithms in terms of experimentally achieved competitive ratio. The figure shows the corresponding performance ratio (OPT/ON) for each algorithm. As competitive ratio measures the performance of algorithm against optimum offline algorithms, thus, a value close to 1 shows the better performance of algorithm. For all trading periods, YFKT performance remains the best in terms of competitive ratio. DAS and BH are among the worst performing algorithms.

Number of Transactions: Table 3 summarizes the number of transactions carried out by each algorithm on $DAX30$ dataset. It can be seen that irrespective of the length of trading periods, non-preemptive algorithms RPMm, KT and BH have only two transactions per trading period. This is because the working of non-preemptive algorithm which invests the wealth (for both buy and sell) at one point, thus one buy and sell transaction in a trading period. In preemptive algorithms, the number of transactions varies and depends on the length of investment horizon. Among preemptive algorithms, YFKT has the least number of transactions, which varies from 4.36 (trading period of length 10) to 25.7 (trading period of length 260), whereas the maximum number of transactions are recorded for DAS, which amounts to the number of trading days in the investment horizon.

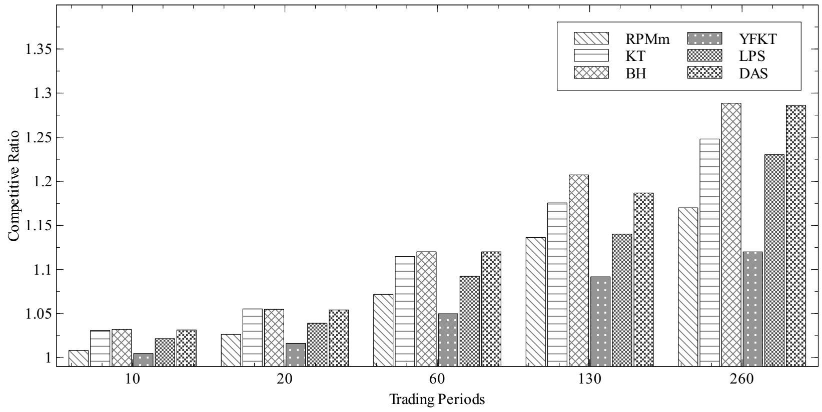


Figure 1 Experimentally achieved competitive ratio on DAX30.

Table 3 Average number of transactions per period.

Period	10	20	60	130	260
RPMm	2	2	2	2	2
KT	2	2	2	2	2
BH	2	2	2	2	2
YFKT	4.36	6.272	11.3	18.45	25.7
LPS	5.98	10.96	32.7	67.15	131.3
DAS	10	20	60	130	254.2

5 Discussion

Based on the different evaluation criteria and the resultant performance, we observe that the YFKT is the best among set of considered algorithms. The performance gap between YFKT and the rest of the considered set of algorithms can be summarized by the fact that YFKT remains the best performing algorithm for trading periods of all lengths. Considering average period return (APR) the performance of YFKT is found on average 2% to 11% better over a trading period of length 10 to 260. Another significant observation is the performance of RPMm, which was found to be the second best. The performance difference in term of RPMm and YFKT, considering APR is on average 3.02%, with a minimum difference of 0.19% (trading period of length 10) and maximum difference of 6.76% (trading period of length 260).

BH and DAS are the two least performing algorithms considering APR as evaluating criterion. On $DAX30$, BH returns are negative (less than 1) for all trading periods. While considering AGR as performance evaluation criterion, there is no significant change in the performance ordering of the algorithms, YFKT and RPMm are the best performing whereas BH and DAS are the least performing algorithms.

The performance comparison of algorithm based on AGR and APR reflects that investment horizon of smaller length results in over all higher returns as the accumulated wealth after each trading period is invested over and over again in the next trading period. For instance, if we consider the AGR of YFKT for trading period of length 10, YFKT returns are 1.4411 in comparison to the APR of same trading period which is 1.0142 only.

Analyzing performance of algorithms, the better performance of YFKT can be attributed

Table 4 Gap between theory and practice.

Algorithm	r_e	r_w	**Gap** (r_e/r_w)
RPMm	1.0104	0.795	1.271
BH	0.8204	0.5573	1.472
YFKT	1.0694	0.878	1.218
LPS	1.1249	1.0975	1.024
DAS	0.81	0.5192	1.559

to the underlying principles and assumptions of the algorithm, YFKT assumes information about the lower and upper bound of offered prices and it neither invests at one point of time nor on all days but invests a portion of wealth when it encounters a new maximum (minimum for buying). Thus, it results in better performance in comparison to other algorithms which either converts at single price based on some pre-calculated reservation price or converts on all days irrespective of the offered price. Although, LPS works on the same principle as YFKT, i.e., convert only when a new maximum (minimum for buying) is encountered, it is not as competitive as YFKT, the main reason can be traced to the amount of wealth invested (s_t) after an investment decision is made. YFKT considers the offered price q_t when calculating s_t whereas LPS does not take into account the offered price q_t, but instead invests an equal portion of remaining wealth based on the remaining numbers of days.

In terms of number of transactions, all non-preemptive algorithms (RPMm, KT, BH) carries 2 transactions in each trading period, one each for buying and selling, whereas the number of transactions performed by preemptive algorithms (YFKT, LPS, DAS) varies depending on the length of trading period. In comparison to other preemptive algorithms, YFKT performs the least transactions. On dataset $DAX30$, the average number of transactions varies from 4 to 25 for trading period of lengths 10 to 260. The highest number of transactions are performed by DAS which is equal to the number of days in the investment horizon.

The relatively low number of transactions of YFKT can be attributed to the working principle of the algorithm, as it does not convert on every offered price but considers progressively higher (lower for buying) prices only, this not only results in better performance of algorithm but also reduces the number of transactions. DAS has the largest number of transaction, one each per day, whereas LPS which also invests only on the highest (lowest for buying) price seen, resulting in lower number of transactions than DAS.

Gap between theory and practice: Analyzing the gap between the theoretical worst case and experimental observed performance of an algorithm is an important aspect of the experimental evaluation of algorithms. We consider yearly data (i.e., dataset of length 260 only and exclude the shorter trading periods) to observe the gap between the theoretical worst case and experimentally achieved performance. For each yearly data, we record the (possible) worst case return (r_w) of the algorithm as well as the experimentally observed return (r_e) and select where the gap (ratio of worst case to that of observed performance) is the highest. For example to calculate worst case return (r_w) of RPMm, we assume that the algorithm achieves the worst case competitive ratio in both buy and sell periods (i.e., RPMm buys at $q_b = \sqrt{Mm}$, whereas OPT buys at m, RPMm sells at $q_s = \sqrt{Mm}$ and OPT sells at M. Thus the worst case return of RPMm is q_s/q_b). The algorithm suggested by Kao and Tate [3] is not included as the working of the algorithm is based on "rank" rather than "actual prices". (Please see [3] Theorem 2.1). It is pertinent to note that we selected only the instances where the gap between the worst case and experimentally observed return is the maximum.

Table 4 summarizes the gap between theoretically worst case and experimentally observed returns of algorithms. It can be seen that DAS and BH has a considerable gap between the worst case and experimentally achieved returns. The least r_e to r_w ratio is observed for LPS, whereas RPMm and YFKT have almost identical gaps.

6 Conclusion

We present an experimental study of online algorithms for the trading problem and compare the results on real world data with an optimum offline algorithm and BH. We observe that all online algorithms perform better than BH and that performance of YFKT is the best among the considered set of algorithms on the $DAX30$ dataset. Further, we deduce that performance behavior of algorithms depends on two main factors, the extent of a-priori information available to the algorithm and the amount of wealth invested per transaction.

The study also finds a number of open questions, such as the performance of YFKT and RPMm is based on the a-priori information such as m and M, however, in real world the a-priori information is subject to errors, it will be interesting to note the performance degradation if the a-priori information is erroneous.

References

1 W.A. Brock, J. Lakonishok, and B. LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance*, 47:1731–1764, 1992.

2 R. El-Yaniv, A. Fiat, R.M. Karp, and G. Turpin. Optimal search and one-way trading algorithm. *Algorithmica*, 30:101–139, 2001.

3 K.-Y. Kao and S.R. Tate. On-line difference maximization. *SIAM Journal on Discrete Mathematics*, 12(1):78–90, 1999.

4 K.-Y. Kwon and R.J. Kish. A comparative study of technical trading strategies and return predictability: An extension of brock, lakonishok, and lebaron (1992) using NYSE and NASDAQ indices. *Quarterly Review of Economics and Finance*, 42(3):613–633, 2002.

5 J. Lorenz, K. Panagiotou, and A. Steger. Optimal algorithms for k-search with application in option pricing. *Algorithmica*, 55(2):311–328, 2009.

6 T.C. Mills. Technical analysis and the london stock exchange: testing trading rules using the FT30. *International Journal of Finance & Economics*, 2(4):319–331, 1998.

7 E. Mohr and G. Schmidt. Trading in financial markets with online algorithms. In B. Fleischmann, K.-H. Borgwardt, R. Klein, and A. Tuma, editors, *Operations Research Proceedings 2008*, pages 33–38. Selected Papers of the Annual International Conference of the German Operations Research Society (GOR) University of Augsburg, September 3-5, 2008, Springer, Heidelberg, 2008.

8 M. Ratner and R. P.C. Leal. Tests of technical trading strategies in the emerging equity markets of latin america and asia. *Journal of Banking and Finance*, 23:1887–1905, 1999.

9 G. Schmidt, E. Mohr, and M. Kersch. Experimental analysis of an online trading algorithm. *Electronic Notes in Discrete Mathematics*, 36:519–526, 2010.

10 P. Shen. Market-timing strategies that worked. *Jounal of Portfolio Management*, 29(2):57–68, 2003.

11 B.M. Tabak and E.J.A. Lima. Market efficiency of brazilian exchange rate: Evidence from variance ratio statistics and technical trading rules. *European Journal of Operational Research*, 194:814–820, 2009.

An exact algorithm for the uncertain version of parallel machines scheduling problem with the total completion time criterion

Marcin Siepak[1]

1 **Wroclaw University of Technology/Institute of Informatics**
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw
`Marcin.Siepak@pwr.wroc.pl`

Abstract

An uncertain version of parallel and identical machines scheduling problem with total completion time criterion is considered. It is assumed that the execution times of tasks are not known a priori but they belong to the intervals of known bounds. The absolute regret based approach for coping with such an uncertainty is applied. This problem is known to be NP-hard and a branch and bound algorithm (B&B) for finding the exact solution is developed. The results of computational experiments show that for the tested instances of the uncertain problem — B&B works significantly faster than the exact procedure based on enumeration of all the solutions. The algorithm proposed has application for further research of quality evaluation for heuristic and approximate solution approaches for the considered problem (in order to check how far from the optimality are solutions generated by them) and also in the cases where the requirement is to obtain the exact solutions for the uncertain problem.

1998 ACM Subject Classification I.2.8 Problem Solving, Control Methods, and Search

Keywords and phrases Task scheduling, Interval uncertainty, Minmax regret, Branch and Bound

Digital Object Identifier 10.4230/OASIcs.SCOR.2012.53

1 Introduction

$P\|\sum C_i$ is classical scheduling problem polynomially solvable ([3], [5]). It consists in scheduling the set of I tasks on the set of J parallel and identical machines. The execution times p_i, $i = 1, \ldots, I$ are given. The optimal schedule minimizes the total flow time $\sum_i C_i$, i.e. the sum of tasks completion times where C_i is the completion time of the ith task. Such a schedule, at each point of time maximizes the number of tasks already processed, by first scheduling those which execution time is the shortest. In order to solve the problem, it is required to assign tasks to the machines and also fix the tasks execution order on each machine.

In this paper we consider the uncertain version of $P\|\sum C_i$ where the execution times p_i are imprecise. The uncertainty is modelled by the concept of a *scenario* which is an assignment of possible values into the imprecise parameters of the problem [1]. The set of all possible scenarios can be described in two ways — in the discrete scenario case, the possible values of uncertain parameters are presented explicitely and in the interval scenario case they can take any value between a lower and upper bound. Hereinafter we consider the interval case where we assume that for each of tasks only the borders of the intervals $\underline{p}_i$ and $\overline{p}_i$ are given where $\underline{p}_i \leqslant \overline{p}_i$ and $p_i \in \left[\underline{p}_i, \overline{p}_i\right]$. Such a way of uncertainty description is useful in the cases where no historical data is available regarding the imprecise parameters, which

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn; pp. 53–63
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

would be required in order to obtain the probability distribution and apply the stochastic approach and also when there is lack of experts opinions which would be a source of other representations of uncertain execution times, e.g. in the form of a membership function for the fuzzy approach.

To solve the nondeterministic case, one can assume the specific values of uncertain parameters and use the deterministic approach. The quality of such a solution however, may be poor because assumed values may differ greatly from the actual ones. For the considered nondeterministic problem, we use therefore the approach based on the *absolute opportunity loss* (*regret*) introduced by Savage in [12] and the concept of *minmax regret* ([2], [9]) which requires finding a feasible solution that is ϵ-optimal for any possible scenario with ϵ as small as possible [1].

Minmax regret versions of many classical optimization problems were studied in [1] and [6]. Newer results were recently obtained in [2], [7], [13], [14].

The paper is organised as follows. The deterministic and uncertain versions of the considered problem are formulated in Section 2. The solution algorithm proposed is presented in Section 3. Section 4 is devoted to description of the computational experiments and their results. In Section 5, the conclusions are presented.

2 Problem formulation

2.1 Deterministic case

Let us introduce the following notation:
$\boldsymbol{I} = \{1, 2, \dots, I\}$ — set of tasks,
$\boldsymbol{J} = \{1, 2, \dots, J\}$ — set of machines,
$p = [p_i]_{i=1,\dots,I}$ — vector of task execution times.
Moreover, let $x = [x_{ikj}]_{i=1,\dots,I;k=1,\dots,I;j=1,\dots,J}$ be the matrix of binary optimization variables where $x_{ikj} = 1$ if the ith task is scheduled as the kth to the last task on machine j, and 0 otherwise. Each machine has therefore I virtual positions, where tasks can be assigned to and parameter k specifies index of the position to the last where a task can be performed.

The objective function is expressed as the sum of tasks completion times [11]:

$$F(p, x) = \sum_{i=1}^{I} p_i \sum_{j=1}^{J} \sum_{k=1}^{I} k x_{ikj}. \tag{1}$$

The following constraints are imposed on optimization variables x_{ikj}. Each task is performed on exactly one position of exactly one machine:

$$\sum_{j=1}^{J} \sum_{k=1}^{I} x_{ikj} = 1; \;\; i = 1, \dots, I. \tag{2}$$

Maximally one task can be performed on each position of each machine:

$$\sum_{i=1}^{I} x_{ikj} \leqslant 1; \;\; j = 1, \dots, J; \;\; k = 1, \dots, I. \tag{3}$$

The elements of matrix x are binary optimization variables:

$$x_{ikj} \in \{0, 1\}\,; \;\; i = 1, \dots, I; \;\; j = 1, \dots, J; \;\; k = 1, \dots, I. \tag{4}$$

Therefore, formulation of deterministic version of $P\|\sum C_i$ is as follows

$$F^{'}(p) \triangleq F(p,x^{'}) = \min_{x} F(p,x) = \sum_{i=1}^{I} p_i \sum_{j=1}^{J} \sum_{k=1}^{I} k x^{'}_{ikj} \tag{5}$$

subject to (2), (3) and (4).

2.2 Uncertain case

For the uncertain case, we assume that the exact value of execution time p_i is not given, and this parameter belongs to the interval $\left[\underline{p}_i, \overline{p}_i\right]$ where $\underline{p}_i$ and $\overline{p}_i$ are known. It means that we consider uncertain parameters p_i described by a set of their possible values in the form of intervals. No other characteristics of such an uncertainty are assumed or used. A particular vector p expresses fixed configuration of the execution times and is called a scenario. A set

$$\boldsymbol{P} = \left[\underline{p}_1, \overline{p}_1\right] \times \ldots \times \left[\underline{p}_I, \overline{p}_I\right] \tag{6}$$

of all scenarios is characterized by the Cartesian product of all the intervals. A scenario where completion times of all tasks are equal borders of the corresponding intervals $\underline{p}_i$ or $\overline{p}_i$ is called an *extreme scenario*.

In order to measure solution quality for the nondeterministic problem, we apply absolute regret as introduced by Savage in [12]. It expresses the difference between value of the total flow time criterion for given solution x and specified scenario p as well as optimal value of the total flow time criterion for p:

$$F(p,x) - F^{'}(p). \tag{7}$$

To find solution of the problem, the robust approach ([1], [9]) is applied. For specified solution x the uncertain parameters are determined by obtaining so called *worst case scenario* denoted by p^x, i.e. the one which maximizes absolute regret:

$$z(x) = \max_{p \in \boldsymbol{P}} \left[F(p,x) - F^{'}(p)\right]. \tag{8}$$

The optimal solution x^* for the uncertain version of $P\|\sum C_i$ minimizes (8), i.e. $z^* \triangleq z(x^*) = \min_x z(x)$ subject to (2), (3) and (4).

3 Solution algorithm

3.1 Deterministic case

The deterministic version of $P\|\sum C_i$ is polynomially solvable and the optimal solution can be obtained using SPT (Shortest Processing Time) rule [11]. This procedure sorts all the tasks in nonincreasing order according to their execution times. They are assigned iteratively then, i.e. the i'th task is assigned into the first available position to the last (i.e. not occupied position, such that the value of parameter k is the smallest) on the n'th machine, where

$$n = \begin{cases} J, & \text{when } (i \bmod J) = 0 \\ (i \bmod J), & \text{otherwise.} \end{cases} \tag{9}$$

3.2 Uncertain case

The uncertain version of $P\|\sum C_i$ is NP-hard which results directly from NP-hardness of the nondeterministic version of a single machine scheduling problem, i.e. $1\|\sum C_i$ [10].

Let us denote by $\boldsymbol{P}_e$ the set of extreme scenarios, i.e.

$$\boldsymbol{P}_e = \left\{\underline{p}_1, \overline{p}_1\right\} \times \ldots \times \left\{\underline{p}_I, \overline{p}_I\right\}. \tag{10}$$

and by p^x the worst-case scenario for solution x.

▶ **Lemma 1.** *$p^x \in \mathbf{P}_e$ for any feasible solution x.*

Proof. It is easy to see, that for two feasible solutions x^1 and x^2 the following equality holds:

$$\begin{aligned} F(p,x^1) - F(p,x^2) &= \sum_{i=1}^{I} p_i \sum_{j=1}^{J}\sum_{k=1}^{I} k x^1_{ikj} - \sum_{i=1}^{I} p_i \sum_{j=1}^{J}\sum_{k=1}^{I} k x^2_{ikj} \\ &= \sum_{i=1}^{I} p_i \sum_{j=1}^{J} \left(k_{ij}^{x^1} - k_{ij}^{x^2}\right) \end{aligned} \tag{11}$$

where for any feasible solution x^s:

$$k_{ij}^{x^s} = \begin{cases} k \in \{1,\ldots,J\} & \text{if } \exists k : x^s_{ikj} = 1 \\ 0 & \text{if } \forall k = 1,\ldots,I \ x^s_{ikj} = 0. \end{cases} \tag{12}$$

As a result of (11) we get:

$$F(p,x) - F^{'}(p) = \sum_{i=1}^{I} p_i \sum_{j=1}^{J} \left(k_{ij}^{x} - k_{ij}^{x^{'}}\right). \tag{13}$$

Now, in order to maximize (13) and obtain the worst-case scenario, it is enough to assume:

$$p_i = \begin{cases} \overline{p}_i, & \text{when } \exists j_x, j_{x'} \in \{1,\ldots,J\} : \ k^x_{ij_x} > 0 \ \wedge \ k^{x^{'}}_{ij_{x'}} > 0 \ \wedge \ k^x_{ij_x} \geqslant k^{x^{'}}_{ij_{x'}} \\ \underline{p}_i, & \text{otherwise.} \end{cases} \tag{14}$$

From (14) it results immediately, that $\forall i$ either $\overline{p}_i \in p^x$ or $\underline{p}_i \in p^x$ and consequently $p^x \subset \boldsymbol{P}_e$ for each solution x. ◀

3.3 Branch and bound algorithm

As a consequence of NP-hardness of the uncertain version of $P\|\sum C_i$, we apply Branch and Bound procedure (B&B) [4] which allows to obtain the exact solution for small instances of the problem faster than the algorithm based on a simple enumeration (denoted as EXCT). B&B can therefore be applied to larger instances of the uncertain problem, which solving using EXCT procedure would take unacceptably long time in the real life systems. According to B&B, the main complex and hard to solve input problem denoted as OPT is decomposed into a finite number of subproblems $OPT_1, OPT_2, ..., OPT_q$ where the corresponding subsets of feasible solutions $Q_1, \ldots, Q_q$, fulfill conditions $Q_l \cap Q_m = \emptyset$ for $l \neq m$ and $Q_1 \cup \ldots \cup Q_q = Q$, where Q is a set of feasible solutions for OPT. The process of decomposition can represented in a form of partition tree where the root corresponds to the input problem OPT and the nodes represent individual subproblems. For each subproblem, until it is fulfilled by maximally one solution — a further decomposition can be performed.

B&B remembers in each iteration the best solution $\acute{x}$ found so far and the corresponding value of the objective function $z(\acute{x})$ or its upper bound $z_{\mathrm{UB}}(\acute{x})$. This algorithm also requires knowledge of the feasible initial solution which can be obtained by applying any approximate or heuristic algorithm. For the purpose of considered uncertain problem, we obtain it by assuming the execution times are equal middles of the corresponding intervals and solving the deterministic version of the problem.

Let $\hat{x}_l$ be the optimal solution for OPT_l. Solving the subproblem directly can still be a process of high complexity, therefore instead of finding $\hat{x}_l$ — a relaxation of OPT_l is performed and optimal solution $\tilde{x}_l$ for the relaxed subproblem is sought. The relaxation generally consists in decreasing subproblem's complexity by weakening some of its restrictions and making it easier to solve. As a consequence of relaxation — solution $\tilde{x}_l$ may not necessirely be feasible for OPT_l, however it fulfills condition $z(\tilde{x}_l) \le z(\hat{x}_l)$, therefore it can be treated as a lower bound for the subproblem, i.e. $z(\tilde{x}_l) = z_{\mathrm{LB}}(\hat{x}_l)$.

Listing 1: Partition tree browse procedure

```
Generate subproblem CurrentSubproblem at the first level of partition tree by
assigning the first task into the last position of the first machine;
repeat
    if CurrentSubproblem is not closed then
        // If subproblem is not closed - then we generate a child node.
        CurrentSubproblem ← GenerateChild(CurrentSubproblem);
        i ← i + 1;
        x́ ← TryToCloseSubproblem(CurrentSubproblem, i, I, x́);
    else
        // If subproblem is closed - we try to generate its neighbour
           then.
        NeighbourSubproblem ← GenerateNeighbour(CurrentSubproblem);
        if NeighbourSubproblem was created then
            CurrentSubproblem ← NeighbourSubproblem;
            x́ ← TryToCloseSubproblem(CurrentSubproblem, i, I, x́);
        else
            CurrentSubproblem ← ParentSolution;
            i ← i − 1;
            Close CurrentSubproblem ;
        end
    end
until CurrentSubproblem is root node;
```

All the subproblems are generated dynamically during execution of B&B, so their total number denoted as q is known after execution of the algorithm is completed. In order to solve OPT, it is required to close each of generated subproblems. The closure of a subproblem implies that that it will not be decomposed anymore, as doing so — would not lead to finding the optimal solution. The individual subproblem OPT_l can be closed when any of the following conditions is fulfilled:

(a) No feasible solution exists that fulfills OPT_l.
(b) Any of conditions $z_{\mathrm{LB}}(\hat{x}_l) > z(\acute{x})$ or $z_{\mathrm{LB}}(\hat{x}_l) > z_{\mathrm{UB}}(\acute{x})$ hold, so as a result optimal solution of OPT_l cannot be optimal for OPT.
(c) A solution $\tilde{x}_l$ is found which is feasible for the non relaxed version of OPT_l.

If none of the above conditions occurs, then OPT_l must be decomposed into further subproblems. Closing the subproblem based on condition (c) when additionally $z(\tilde{x}_l) < z(\acute{x})$ occurs, means that a new best solution was found, therefore substitution $\acute{x} = \tilde{x}_l$ can be performed.

Each node of the partition tree specifies unambiguously the partial allocation matrix $\tilde{x}$, which at the last, I'th level indicates the complete solution x of the scheduling problem. At the i'th level of the partition tree ($i = 1, \ldots, I$), $i - 1$ tasks have already been assigned to the machines and the allocation of the i'th task is performed, while $(I - i)$ tasks will be assigned on the next levels $i + 1, i + 2, \ldots$ of the tree (unless the subproblem can be closed before that). The procedure presented on Listing 1 generates and browses the partition tree. The following notation is used: *CurrentSubproblem* — subproblem corresponding to the currently processed node, i — level of partition tree corresponding to *CurrentSubproblem*, I — total number of tasks.

According to the procedure presented on Listing 1 — if a current subproblem could not be closed (Line 3), then we decompose it (*GenerateChild* procedure — Line 4). Otherwise — we try to generate its neighbour (*GenerateNeighbour* procedure — Line 8), and if that succeds, we assign neighbour as the current node and try to close it (*TryToCloseSubproblem* — Line 6).

The procedure *GenerateChild* applied for *CurrentSubproblem* at the i'th level of partition tree generates the first child only which is allocated at level $(i + 1)$ and the attempt to close it is performed immediately after it has been created (Line 6). If closure fails to succeed, then *GenerateChild* procedure is applied in the next iteration (Line 4), in order to generate child node at level $(i + 2)$. Otherwise — if it was closed successfully, then its neighbour is tried to be generated in the next iteration using *GenerateNeighbour* procedure (Line 8). Each execution of *GenerateNeighbour* applied for any node generates at most one its direct neighbour or returns empty value when no neighbour node could be generated. Returning empty value causes that the procedure goes to the parent node which is automatically closed then (Line 15) and its neighbour is tried to be generated in the next iteration of the algorithm (Line 8). The existence of neighbour node is determined by the analysis of *CurrentSubproblem* to which *GenerateNeighbour* procedure is applied.

Listing 2: TryToCloseSubproblem procedure

Input : *CurrentSubproblem*, i, I, $\acute{x}$
Output : $\acute{x}$

Find solution $\tilde{x}$ for the relaxed version of *CurrentSubproblem*;
if $z_{\text{LB}}(\tilde{x}) \geq z_{\text{UB}}(\acute{x})$ **then**
    Close *CurrentSubproblem* ; `// We close the current subproblem`
else
    if $i == I$ **then**
        `// That means` $\tilde{x}$ `is a feasible solution for the nonrelaxed subproblem`
        Close *CurrentSubproblem*;
        if $z_{\text{UB}}(\tilde{x}) < z_{\text{UB}}(\acute{x})$ **then**
            $\acute{x} \leftarrow \tilde{x}$; `//` $\tilde{x}$ `is the best solution we currently have`
        end
    end
end
return $\acute{x}$

The subprocedure $TryToCloseSubproblem$ presented in Listing 2 evaluates the partial solution related to the current node and checks if the subproblem related to it can be closed, so no more children of that node would be generated.

The child node is generated according to the following property which improves solution quality and is true for each optimal solution [11]:

Property: If job i is assigned to position $k > 1$ on machine j, then there is also a job assigned to position $k-1$ on the machine j. Otherwise scheduling job i on position $k-1$ would improve the total assignment cost.

While generating child at i'th level of the tree (*GenerateChild* procedure), we try to assign task $i+1$ into the latest available position (i.e. the one where value of corresponding parameter k is the smallest, as parameter k specifies the position to the last) of the first available machine. The indexes of available machines and positions depend however on the assignment of tasks $1,\dots,i$ on the corresponding previous levels of the partition tree. Let us denote by k_j the value of k for the earliest position occupied (by any task) on machine j. That means, k_j equals the highest value of k corresponding to position occupied on machine j in the current partial solution. As an example, let's assume that $I = 10$, $J = 2$ and $k_2 = 4$. So, the task currently assigned as the earliest on machine 2 is performed as the 4'th to the last. Let us denote by $kMax_j$ the earliest possible position where the current task can be assigned to on machine j in order to fulfill the above property. In case of the presented example, $kMax_1$ must equal 6, which means that current task can be performed on machine 1 the earliest at position 6'th to the last. Assigning this task any earlier on machine 1 would cause unnecessary increase of the total completion time criterion (1), as the property presented above would not be fulfilled then. Therefore, while performing *GenerateChild* procedure, first we determine parameters $k_j, (j = 1, ..., J)$ for each machine. Then we try to assign task $i+1$ into machine j, starting from $j = 1$. We iteratively analyse position k'th to the last on the j'th machine within range $k = 1,\dots,kMax_j$ and check if it is available. If the position is free, then we assign there the current task, and *GenerateChild* procedure stops. Otherwise we try to perform the assignment of task $(i+1)$ on machine $(j+1)$. Parameter $kMax_j$ is calculated according to the following formula:

$$kMax_j = \max\{k_j, I - k_1 - \ldots - k_{j-1} - k_{j+1} - \ldots - k_J\}. \tag{15}$$

While building neighbour node for the *CurrentSubproblem* at the i'th level of the tree we use the knowledge of indexes for machine and position of where the i'th task has been assigned in *CurrentSubproblem*. Let us denote these indexes as $jCurr$ and $kCurr$ respectively. The procedure starts from trying to assign the task into the next available (i.e. not occupied) position on machine $jCurr$ where parameter k corresponding to that position is greater than $kCurr$. So, the i'th task is tried to be assigned iteratively on machine $jCurr$ into the position earlier than it has already been assigned on this machine in the node corresponding to *CurrentSubproblem*. If all the positions up to the one where $k = kMax_{jCurr}$ were occupied, then the assignment is tried to be performed starting from the last position (i.e. $k = 1$) of the next machine $(jCurr + 1)$ until $k = kMax_{jCurr+1}$. The procedure stops, when the task assignment was successfully made or when position $kMax_J$ on the J'th machine was processed and no assignment could be performed due to the occupancy of all the processed positions (the empty value is returned then).

3.4 Lower bound calculation

Having given partial solution $\dot{x}$ corresponding to a specified node at the ith level of the partition tree, calculation of the lower bound z_{LB} is required for this subproblem. In order to perform it, we start from determining parameters $k_j, j = 1, \ldots, J$ for each machine (similarly like in case of *GenerateChild* procedure) and generate series of finite sequences $m_j = (m_{j_s})$ where each sequence contains values of parameter k of those positions on machine j which have not been occupied by any task yet (starting from $k = 1$ up to k_j). That means, each element of m_j specifies index of a position to the last on machine j, where one of non assigned yet tasks will need to be allocated to (in order to fulfill the property presented in Section 3.3) — and as a result its execution time will be multiplied by the value of k corresponding to that position. Let $\overline{m}_j$ denote the number of elements of m_j and m be a sequence of pairs $(m_{j_s}, j), s = 1, \ldots, \overline{m}_j; j = 1, \ldots, J$ concatenating elements of sequences m_j. In order to calculate the lower bound, we generate all the possible combinations of size $\overline{m}$ ($\overline{m}$ is the length of m) from these tasks which have not been assigned to any machines yet, i.e. tasks $i + 1, \ldots, I$. Then, for each combination set, denoted as $\boldsymbol{N}$ we generate all the permutations of $\boldsymbol{N}$. Let $n = (n_s), s = 1, \ldots, \overline{m}$ be a single permutation sequence. Now, we modify $\dot{x}$ and allocate task n_s into the machine and position to the last specified by element s of sequence m ($s = 1, \ldots, \overline{m}$). For $\dot{x}$ modified in such a way we calculate the value of absolute regret. Then we start again from the initial partial solution $\dot{x}$ and modify it using the next permutation sequence calculating absolute regret right after the modification is done. Such a process is repeated until all the permutation sequences for all the combination sets have been analysed. The smallest calculated value of absolute regret is the lower bound for the relaxed version of subproblem and the corresponding $\dot{x}$ is the optimal solution for the relaxed subproblem (we denote such a solution as $\tilde{x}$).

In order to calculate absolute regret (7) for $\dot{x}$ — knowledge of the worst case scenario $p^{\dot{x}}$ is required. The occurrence of parameter k in every addend of (1) causes however, that finding $p^{\dot{x}}$ is NP-hard problem, as it requires analysis of all the extreme scenarios, which number is $2^{|P_e|}$. Therefore, Listing 3 presents how to efficiently generate $\widehat{p}^{\dot{x}}$ which is approximation of unable to calculate effectively scenario $p^{\dot{x}}$ and as a result — how to obtain lower bound of absolute regret.

Listing 3: Calculation of the lower bound for absolute regret

Input : $\dot{x}$
Output : $\tilde{z}$— lower bound for absolute regret z

Obtain scenario $\overline{p} = [\overline{p}_i]_{i=1\ldots,I}$;
for $i \leftarrow 1$ **to** I **do**
    Generate scenario $\tilde{p} \leftarrow \overline{p}$;
    Assign $\tilde{p}_i \leftarrow \underline{p}_i$ in scenario $\tilde{p}$;
    // $\tilde{z}(p,\dot{x})=F(p, \dot{x}) - F^{'}(p)$ for any p
    if $\tilde{z}(\tilde{p},\dot{x})>\tilde{z}(\overline{p},\dot{x})$ **then**
        $\overline{p} \leftarrow \tilde{p}$;
    end
end
return $\tilde{z}(\overline{p},\dot{x})$

3.5 Upper bound calculation

For the initial solution generated assuming execution times equal middles of the intervals and also for each solution x feasible for the input problem OPT (e.g. the one generated at the leaf of partition tree), calculation of the upper bound z_{UB} is necessary. In order to perform it, we propose to find sets P_m and P_s of scenarios fulfilling the following conditions

$\forall p_m \in P_m : \; F(p_m, x) \geq F(p^x, x)$ and $\forall p_s \in P_s : \; F'(p_s) \leq F'(p^x)$. The value of z_{UB} is then calculated using the following formula:

$$z_{\mathrm{UB}}(x) = \min_{p_m, p_s} \left[F(p_m, x) - F'(p_s) \right]. \tag{16}$$

In order to build set P_s, we apply the procedure presented on Listing 4:

Listing 4: Generation of set P_s

1 $z_{\mathsf{max}} \leftarrow 0$;
2 Generate scenario $\underline{p} \leftarrow \left[\underline{p}_i\right]_{i=1\ldots,I}$;
3 **for** $i \leftarrow 1$ **to** $\lfloor I/2 \rfloor$ **do**
4   **repeat**
5   Get next scenario p_i by modifying i execution times in $\underline{p}$ from the lower to the upper bounds of the corresponding intervals;
          // $\tilde{z}(p,x)$=$F(p,x) - F'(p)$ for any p
6   **if** $\tilde{z}(p_i,x) \geq z_{\mathsf{max}}$ **then**
7   **if** $\tilde{z}(p_i,x) > z_{\mathsf{max}}$ **then**
8   Clear set P_s;
9                 $z_{\mathsf{max}} \leftarrow \tilde{z}(p_i,x)$;
10   **end**
11   Add p_i to set P_s;
12   **end**
13   **until** *all possible scenarios p_i have been processed*;
14 **end**

The procedure of finding set P_m follows the one as on Listing 4 with the difference, that scenario $\overline{p} = [\overline{p}_i]_{i=1\ldots,I}$ is used instead of $\underline{p}$ (Line 2) and while performing each iteration, scenarios having i execution times equal lower bounds of the corresponding intervals (Line 5) are processed ($i = 1, \ldots, \lfloor I/2 \rfloor$). After obtaining sets P_s and P_m, the upper bound is calculated according to (16).

4 Computational experiments

All the algorithms have been written in c# and performed on Intel Core i5, 2.40GHz, 4.00 GB of RAM. In order to test the quality of developed B&B, we compared it with the exact algorithm based on enumeration of all the possible solutions. The test instances of the uncertain problem were generated according to the procedure presented in [8]. First we introduced parameter $C = \{10, 50, 100\}$. For each C the experiments were performed separately. Starting from $C = 10$, for the i'th task ($i = 1, \ldots, I$) we generated $\underline{p}_i$ randomly from the interval $[1, C]$ and $\overline{p}_i$ randomly from the interval $\left[\underline{p}_i, \underline{p}_i + C\right]$ according to the uniform probability distribution. Parameter C specifies the influence of uncertainty on the problem — the higher its value is, the bigger may be the difference between the lower and the upper bound of the intervals for the imprecise parameters. For each C and each of $I = 5, \ldots, 14$ five instances of the uncertain problem were generated and solved independently.

The results of computational experiments for different values of C and $J = 2$ are presented in Table 1. Column I reports the total number of tasks, *Enum* and *B&B* represent the average execution time (in seconds) of the algorithm based on enumeration and Branch and Bound procedure while running it 5 times for different instances of the uncertain problem of

Table 1 The results of experiments for B&B and enumeration procedure.

$C = 10$				$C = 50$				$C = 100$			
I	Enum	B&B	ADV	I	Enum	B&B	ADV	I	Enum	B&B	ADV
5	0.02	0.01	50%	5	0.06	0.01	83.3%	5	0.1	0.04	60%
6	0.202	0.124	38.6%	6	0.24	0.13	45.8%	6	0.18	0.12	33.3%
7	2.62	1.66	36.6%	7	2.76	1.74	37%	7	2.3	1.6	30.4%
8	24.13	13.7	43.2%	8	23.11	12.52	45.8%	8	28.42	14.53	48.9%
9	170	95	44.1%	9	242	120	50.4%	9	309	174	43.7%
10	1087	536	50.7%	10	1353	521	61.5%	10	1493	667	55.3%
11	4108	1781	56.6%	11	4714	2014	57.3%	11	5014	1935	61.4%
12	24990	9985	60%	12	26043	10148	61%	12	29091	10306	64.6%
13	—	27730	—	13	—	29990	—	13	—	31237	—
14	—	71470	—	14	—	77353	—	14	—	81382	—

the same size. Column ADV represents the percentage execution time advantage of B&B over the exact algorithm.

While performing experiments we solved the uncertain problem consisting of up to 12 tasks using the exact algorithm. For the higher number of tasks, the calculation time took more than 24 hours and we stopped the computations. The Branch and Bound procedure solved the problem consisting of up to 14 tasks. The experiments generally show that both of the algorithms work longer while increasing the value of C. For $I = 12$, B&B works even 64.6% faster than the exact procedure, so as a result we could retrieve the exact solution in approximately 2 hours 52 minutes instead of 8 hours 5 minutes while using the exact procedure. The results also show that the execution time advantage of B&B over the exact algorithm increases while increasing the total number of tasks (I). The above observations are limited of course to the tested instances of the uncertain problem.

5 Conclusions

In this paper we have developed and tested a Branch and Bound algorithm for the uncertain version of $P||\sum C_i$ where the uncertain parameters are expressed in the forms of intervals and only their lower and upper bounds are known. Such a version of the problem is known to be NP-hard. The computational experiments for small instances of the problem show that this procedure finds an optimal solution significantly faster than the exact algorithm based on analysis of all the possible solutions. The algorithm proposed has therefore applications for further research of developing efficient approximate and heuristic procedures — in order to test how far from the optimality are the solutions generated by those procedures. Moreover the methodology of building and browsing the partition tree can be applied while extending developed B&B algorithm to solve the uncertain version of more complex problem than considered in this paper, i.e. $R||\sum C_i$. We also recommend to use this method in the cases when the requirement of obtaining the optimal solution has priority — and is more important than the computation time.

References

1 H. Aissi, C. Bazgan, and D. Vanderpooten. Minmax and minmax regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197:427–438, 2009.

2 I. Averbakh and J. Pereira. Exact and heuristic algorithms for the interval data robust assignment problem. *Computers & Operations Research*, 38:1153–1163, 2011.

3 J. Bruno, E. G. Coffman, Jr., and R. Sethi. Scheduling independent tasks to reduce mean finishing time. *Communications of the ACM*, 17:382–387, 1974.

4 M. J. Brusco and S. Stahl. *Branch-and-bound applications in combinatorial data analysis*. Springer, 2005.

5 W. A. Horn. Minimizing average flow time with parallel machines. *Operations Research*, 21:846–847, 1973.

6 A. Kasperski. *Discrete Optimization with Interval Data: Minmax Regret and Fuzzy Approach (Studies in Fuzziness and Soft Computing)*. Springer, Berlin, Heidelberg, New York, 2008.

7 A. Kasperski, A. Kurpisz, and P. Zielinski. Approximating a two-machine flow shop scheduling under discrete scenario uncertainty. *European Journal of Operational Research*, 217:36–43, 2012.

8 A. Kasperski and P. Zielinski. Minimizing maximal regret in linear assignment problems with interval data. Technical Report 07, Raport Serii PREPRINTY, Wroclaw, 2004.

9 P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, Dortrecht, Boston, London, 1997.

10 V. Lebedev and I. Averbakh. Complexity of minimizing the total flow time with interval data and minmax regret criterion. *Discrete Applied Mathematics*, 154:2167–2177, 2006.

11 M. L. Pinedo. *Scheduling - Theory, Algorithms and Systems*. Springer, New York, 2008.

12 L. J. Savage. The theory of statistical decision. *Journal of the American Statistical Association*, 46:55–67, 1951.

13 M. Siepak and J. Jozefczyk. Minmax regret algorithms for uncertain p||cmax problem with interval processing times. In *Proceedings of 21'st International Conference on Systems Engineering*, Las Vegas, USA, 2011.

14 A. Volgenant and C. W. Duin. Improved polynomial algorithms for robust bottleneck problems with interval data. *Computers & Operations Research*, 37:900–915, 2010.

Product Form of the Inverse Revisited*

Péter Tar[1] and István Maros[1]

1 **Department of Computer Science and Systems Technology**
University of Pannonia, Veszprém, Hungary
tar@dcs.uni-pannon.hu

Abstract

Using the simplex method (SM) is one of the most effective ways of solving large scale real life linear optimization problems. The efficiency of the solver is crucial. The SM is an iterative procedure, where each iteration is defined by a basis of the constraint set. In order to speed up iterations, proper basis handling procedures must be applied.

Two methodologies exist in the state-of-the-art literature, the product form of the inverse (PFI) and lower-upper triangular (LU) factorization. Nowadays the LU method is widely used because 120-150 iterations can be done without the need of refactorization while the PFI can make only about 30-60 iterations without reinversion in order to maintain acceptable numerical accuracy.

In this paper we revisit the PFI and present a new version that can make hundreds or sometimes even few thousands of iterations without losing accuracy. The novelty of our approach is in the processing of the non-triangular part of the basis, based on block-triangularization algorithms. The new PFI performs much better than those found in the literature. The results can shed new light on the usefulness of the PFI.

1998 ACM Subject Classification G.1.6 Optimization

Keywords and phrases Linear optimization, Simplex method, Basis inverse, Product form of the inverse

Digital Object Identifier 10.4230/OASIcs.SCOR.2012.64

1 Introduction

Our research aims to revisit the usability and effectiveness of the product form of the inverse in the simplex method in the light of the technological and algorithmic developments of the past decades. In Section 2 we present a brief literature overview of the simplex method and highlight the issues that play a key role in our investigations. These areas are the basis handling procedures, the property of sparsity and the numerical issues of the solution algorithm. In Section 3 we present our novel approach with a detailed description of the processing of the non-triangular part of the basis. In Section 4 a computational study is given to validate our results. Section 5 contains the conclusions.

* This publication/research has been supported by TÁMOP-4.2.2/B-10/1-2010-0025 project.

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn; pp. 64–74
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 Literature overview

The history of linear optimization started in the early 1950's. The problem was originated by Dantzig [2]. One of the possible formulations of the problem is the standard form

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^T\mathbf{x}, \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$; $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$.

The simplex method gives the global optimum of a linear optimization problem. It also has a modular structure. Thus, loosely connected parts of the algorithm can be investigated separately. The simplex method is an iterative algorithm. The state of a simplex iteration can be described by a basis ($\mathbf{B}$) of the linear equation system. In each iteration one variable leaves the basis while another one enters. The whole process is started from an initial basis and ends with an optimal one if it exists.

2.1 Use of the basis in the simplex method

The efficiency of a simplex implementation can be measured by solution times. The total solution time depends on the number of iterations and the time consumed by one iteration. In our approach we focused on the time taken by an iteration. In each iteration two types of operations are necessary to facilitate a basis change:

$$\boldsymbol{\alpha} = \mathbf{B^{-1}}\mathbf{a}, \quad \text{(or solve } \mathbf{B}\boldsymbol{\alpha} = \mathbf{a} \text{ for } \boldsymbol{\alpha}\text{)}, \tag{1}$$

$$\boldsymbol{\alpha}^T = \mathbf{a^T}\mathbf{B^{-1}}, \quad \text{(or solve } \mathbf{B}^T\boldsymbol{\alpha} = \mathbf{a} \text{ for } \boldsymbol{\alpha}\text{)}, \tag{2}$$

where $\mathbf{B}$ is a basis of matrix $\mathbf{A}$ and $\mathbf{a}$ is some vector appearing in the algorithm. The computational effort needed to carry out these computations is significant. This highlights that the implementation of these two computations has an important influence on the performance of the simplex method. In the literature two major methods exist to perform operations with the basis. One is called the product form of the inverse (PFI), which we investigate in our work, the other is the elimination form (EFI) or lower-upper triangular (LU) factorization.

The PFI was introduced in [3]. The idea of this approach is to store the inverse of the basis as the product of special matrices.

$$\begin{aligned} \mathbf{B^{-1}} &= \mathbf{E_m}\mathbf{E_{m-1}}\dots\mathbf{E_1} \\ \mathbf{B_i^{-1}} &= \mathbf{E_i}\mathbf{B_{i-1}^{-1}}, \text{ for } i = 1, \dots, m, \end{aligned}$$

where $\mathbf{B_0^{-1}} = \mathbf{I}$ and, obviously, $\mathbf{B_m^{-1}} = \mathbf{B^{-1}}$

The $\mathbf{E_i}$ matrices are called elementary transformation matrices (ETMs). They differ from the unit matrix in only one column. During a basis change, this form can be updated by computing a new ETM. If the new basis is $\bar{\mathbf{B}}$, then $\bar{\mathbf{B}}^{\mathbf{-1}} = \mathbf{E}\mathbf{B^{-1}}$.

The basic simplex operations of (1) and (2) are performed recursively:

$$\boldsymbol{\alpha} = \mathbf{E}_m\mathbf{E}_{m-1}\dots\mathbf{E}_1\mathbf{a} \tag{3}$$

$$\boldsymbol{\alpha}^T = \mathbf{a}^T\mathbf{E}_m\mathbf{E}_{m-1}\dots\mathbf{E}_1 \tag{4}$$

(3) is called FTRAN (Forward TRANsformation) while (4) is referred to as BTRAN (Backward TRANsformation).

The LU factorization of the basis in linear programming is presented in [7]. The idea is based on the triangular decomposition of the basis $\mathbf{B} = \mathbf{LU}$, where $\mathbf{L}$ is lower and $\mathbf{U}$ is upper triangular matrix. With this approach the FTRAN and BTRAN operations are computed by forward and backward substitutions. Details are omitted in this paper. The aim of our research was to investigate the PFI. During simplex iterations both forms (PFI and LU) can be updated by special formulas, so inversion (or factorization) is not necessary in every iteration. The update operations are relatively cheap in terms of computational effort.

2.2 Sparsity

Large scale real-life linear optimization problems usually have the property of sparsity. This means that there are very few non-zeros relative to the total size of the coefficient matrix $\mathbf{A}$. Experience shows that the average number of non-zero elements in a column is no more than 10, irrespectively of the size of the problem. During computations a vector can be transformed in such a way that the number of non-zero elements increases. This phenomenon is known as *fill-in*.

Independently of the PFI or LU method, triangular reordering of the basis must be performed during inversion (or factorization). The benefit of triangularization comes from sparse computing. It generally means that during inversion (or factorization) of the basis the number of non-zeros in the resulting inverse (or the LU matrices) can be kept low. In our work we focus on large-scale problems, thus exploiting sparsity is a key issue.

2.3 Numerical issues

During simplex iterations numerical errors (rounding, truncation and cancellation) can occur due to the finite precision of the computing architectures. As a general rule, double precision is used. In course of simplex iterations many additive and multiplicative operations are performed. As the computation carries on, these errors can add up and reach such magnitudes that can ruin the results unless numerical errors are handled properly.

As the basis changes numerical values become less accurate. Thus the vectors (like reduced cost, solution, etc.) computed by FTRAN and BTRAN operations lose accuracy, too. It can be said that the accuracy of the representation of the basis determines the accuracy of the resulting vectors. As such, it has a significant effect on choosing the incoming and outgoing variables. The accuracy of the computations can be controlled by reinversion (or refactorization) of the basis. This operation involves substantial computational effort.

The simplex algorithm is considered numerically stable if the following properties are satisfied during the solution process:

- The obtained solution is optimal.
- The numerical value of the objective value is (not strictly) monotonically improving.
- After reinvesion (or refactorization) the basis remains valid. (Numerical problems can lead to a linearly dependent set of vectors as a basis.)

3 Revisiting the product form of the inverse

Nowadays, the PFI is meant to be less effective than the LU form even if some examples exist for which PFI is superior. The idea of revisiting the product form emerged during the development of a simplex solver. We wanted to know the capabilities of the PFI in the light of recent technological and algorithmic developments. The basic problem with the PFI is that reinversion is needed after 30-60 iterations to maintain acceptable numerical accuracy, while this can go up to 100-200 if the LU form is used [8]. There were some efforts to create good PFI, but numerical problems weren't solved [6, 5].

In our work we focused on building a PFI based method and implementation that can solve real-life problems without reinverting frequently. We have built a system that can solve problems (from the *netlib* collection) using reinversion frequencies (number of iterations without reinverting) up to 300-3000.

We have implemented and investigated triangularization methods available in the literature by studying their strengths and weaknesses and also proposed numerical issues to be considered during the FTRAN and BTRAN operations. We present our approach in the following sections.

3.1 The inversion process

During inversion elementary transformation matrices are generated from the columns of the actual basis. An ETM is the result of a pivoting process on the basis. Inversion in product form can be described as follows.

1. Start with a matrix with known inverse, the identity matrix ($\mathbf{I}$) is a simple choice because $\mathbf{I^{-1}} = \mathbf{I}$.
2. Replace the columns of $\mathbf{I}$ with the columns of the basis, so the updating formula can be used.

The ETM for the update formula in the i^{th} step can be computed as:

$$\mathbf{E} = [\mathbf{e_1}, \ldots, \mathbf{e_{i-1}}, \boldsymbol{\eta}_\mathbf{i}, \mathbf{e_{i+1}}, \ldots, \mathbf{e_m}]$$

using

$$\boldsymbol{\eta}_\mathbf{i} = \left[-\frac{v_i^1}{v_i^p}, \ldots, -\frac{v_i^{p-1}}{v_i^p}, \frac{1}{v_i^p}, -\frac{v_i^{p+1}}{v_i^p}, \ldots, -\frac{v_i^m}{v_i^p}\right]$$
$$\mathbf{v_i} = \mathbf{B_{i-1}^{-1}} \mathbf{b_i},$$

where $\mathbf{b_i}$ is column i of $\mathbf{B}$. In practical implementations, after a new ETM is formed, all the remaining columns are updated by it. In this way, the $\mathbf{v_i}$ vectors are available for the next step, there is no need for further computations. While the inverse of a matrix is unique the product form is not. This flexibility can be utilized to achieve some desirable goals. Column permutations can be done on the basis in order to create a form, which gives a low fill-in during the inversion.

3.2 Triangularization method

During inversion the order that we use to replace the columns of $\mathbf{I}$ should be selected carefully. As the remaining columns are updated, the number of non-zeros in $\mathbf{v_i}$ vectors can increase as a result of possible fill-ins. After an ETM is generated all remaining vectors must be updated if they have non-zero elements in the pivot row of the ETM. The other vectors remain unchanged.

Row and column counts are introduced for the basis to represent the number of non-zeros in the corresponding rows and columns. These counts can be used to identify the triangular parts of the basis. The advantage of the triangular parts is that they can be inverted directly without fill-in and without affecting the remaining part.

Figure 1 demonstrates the non-zero structure of a basis after triangularization.

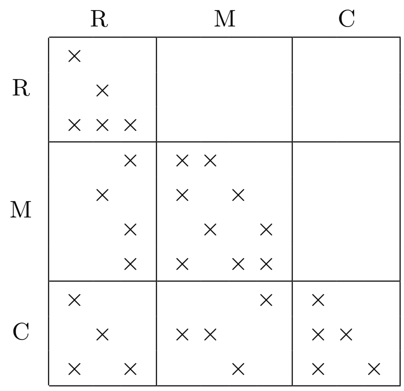


Figure 1 Triangularized form of a basis.

We emphasize that in an efficient implementation the column reordering is done only logically through a permutation vector. First, the R columns are identified and inverted. This can be done without fill-in. Next, C columns are identified. After that, the M columns must be processed (this is the most critical part of the inversion). Finally the C columns are inverted also without fill-in.

3.3 Processing the non-triangular part

The non-triangular part of the basis plays a key role in both the size and the accuracy of the inverse. We have identified three requirements to be satisfied during the processing of this part to obtain a stable inverse.

1. The number of fill-ins must be as low as possible.
2. Beside the number of fill-ins, the number of transformations must also be considered. If there is no fill-in but the non-zero elements of the basis are transformed many times with the update formula then their numerical accuracy can deteriorate. Furthermore, numerical properties must be considered. A well-scaled element can be a better pivot candidate even if it is a result of more transformations.
3. The distribution of the transformations among the non-zeros is important. The accumulation of numerical errors should be prevented or reduced. If the number of transformations is low but there are only a few elements which have been transformed many times then the numerical error accumulated on these specific values can ruin the stability of computations with the inverse.

During inversion all these properties must be taken care of.

To identify some structure in the non-triangular part (which is also sparse), we can use block-triangularization. The Tarjan algorithm [9, 4] is appropriate to identify the block structure. The result is a lower block-triangular form where non-zero elements are permuted into the diagonal. The block-triangular form can reveal hidden triangularity of the basis. This satisfies the first requirement of a stable inverse.

Each block containing more than one column cannot be inverted without numerical transformations. It is important to notice that if pivot elements are chosen within the blocks then numerical transformations must be done only on the columns of the corresponding block. So, numerical errors can not overflow from one block to another. This prevents the adding up of numerical errors throughout the whole non-triangular part thus satisfying the third requirement of a stable inverse.

After running the Tarjan algorithm threshold pivoting can be used to identify the pivot positions within the blocks. This criterion selects a subset from the eligible pivot elements. Threshold pivoting must select a pivot element within a non-triangular block satisfying

$$|v_j^i| \geq u \max\{|v_j^k|\}, \tag{5}$$

where u is an adjustable parameter and v_j is the investigated vector. The value 0.01 for u is appropriate in most cases. From the elements satisfying equation (5) one with the lowest row count is chosen for pivoting, thus the number of necessary updates is kept low.

This criterion hopefully gives an eligible numerical value for the pivot element to maintain accuracy. In special cases it can happen that the only possible choice left in a column is an element of small magnitude. Choosing a too small element as a pivot can result in an η vector with large elements because the elements of the vector $\mathbf{v}$ are divided by the pivot. If the η vector contains large elements then the small numerical errors arising from the finite precision can scale up to the magnitude of "normal" values. In the worst case the only possibility for the algorithm can be to choose a numerical garbage as a pivot element. Both can ruin the inversion procedure and lead to wrong basis changes. As a result of wrong choices after the next reinversion the simplex algorithm usually falls back to phase-1 or the objective value gets much worse.

The threshold pivoting criterion works well if the candidate set contains good elements. Therefore, the pivoting algorithm should take care of this. We have proposed and implemented a heuristic extension for processing the non-triangular part. We consider the number of non-zeros in the columns of a non-triangular block and order the columns with increasing column counts. After that the threshold pivoting procedure is called for the column with the lowest column count. Thus columns having a few non-zeros are pivoted first, preventing them to add up numerical errors on their non-zeros. Columns with higher column counts have numerical advantages in case of threshold pivoting because there is a wider range of non-zeros to be chosen for pivot elements. With this extension the balance between the number of updates and the numerical stability satisfies the second requirement of a stable inverse.

A comprehensive study on the efficiency of the reordering is shown in Section 4.

3.4 Improving numerical stability during computations

Additive operations must be performed during FTRAN, BTRAN and column updating. These operations are the main sources of numerical problems. It can happen that the difference of differently transformed, but algebraically identical quantities is a small number but not zero. In such a case the numerical garbage must be eliminated. We can introduce a relative tolerance (ε_r) and say that if the result of an additive operation is smaller than the absolute value of the larger one multiplied by the tolerance then the result can be considered zero (6). Using this technique improves the numerical properties of the algorithm.

$$a + b = \begin{cases} 0, & \text{if } |a+b| < \varepsilon_r \max\{|a|, |b|\}, \\ a+b, & \text{otherwise.} \end{cases} \tag{6}$$

During additive operations serious cancellation errors can occur. The resulting inaccurate values can scale up significantly in subsequent operations. To reduce the probability of this to happen (e.g., during computing dot products) we collect the positive and negative terms separately and add them up at the end. Similarly, we need to be cautious in case of logical (comparison) operations on two numbers. If equality is tested then absolute tolerance (ε_a) is also used. If the absolute value of the numbers difference is under the tolerance they are considered to be equal.

Both techniques have been implemented and used in our simplex solver. Now we use 10^{-10} for the relative tolerance and 10^{-14} for the absolute tolerance. The pivoting method is extended with a tolerance too, all pivot elements must be over 10^{-6} in absolute value.

4 Computational results

We have performed a computational study of our PFI implementation. For the tests we used the *netlib* collection. In Table 1 we present a subset of these problems which we use to present the findings. The chosen subset contains representatives of the sizeable and numerically more difficult problems. Problems DFL001, QAP12 and QAP15 are omitted because currently our implementation uses the primal algorithm only and these problems can be handled more effectively with the dual.

Table 1 Test problem set from the *netlib* collection.

Problem name	Rows	Columns	Non-zeros	Density
25FV47	822	1571	11127	0.86%
80BAU3B	2263	9799	29063	0.13%
BNL2	2325	3489	16124	0.19%
D2Q06C	2172	5167	35674	0.31%
DEGEN3	1504	1818	26230	0.95%
FIT2D	26	10500	138018	50.55%
FIT2P	3001	13525	60784	0.14%
GREENBEA	2393	5405	31499	0.24%
GROW22	441	946	8318	1.99%
MAROS-R7	3137	9408	151120	0.51%
PILOT	1442	3652	43220	0.82%
PILOT87	2031	4883	73804	0.74%
QAP08	913	1632	8304	0.55%
STOCFOR3	16676	15695	74004	0.02%
TRUSS	1001	8806	36642	0.41%
WOOD1P	245	2594	70216	11.04%

All the following results have been obtained on a personal computer with Intel(R) Core(TM)2 Duo E8400@3.0Ghz, 2GB RAM running 32bit Windows 7. The simplex implementation of our research group is named after the university, it is called Pannon Optimizer (PanOpt). It is important to note that the following results have been reached by using the following settings:

- Presolve techniques are not used.
- Scaling techniques are not used.
- Advanced starting basis finder techniques are not used. Every solution is obtained from the lower logical starting basis.
- Dantzig full pricing is used. This heavily effects the number of iterations needed to obtain the optimal solution.
- All these properties affect the solution time, thus our solution times published are worse than commercial solvers but they can serve as a basis for comparison. The computed optimal solutions have been validated using the COIN-OR CLP software [1].

4.1 Study of the efficiency of column reordering

In this section the numbers of non-zero transformations are analyzed. Two measurements are used for comparison. Both use the lower block-triangular form generated by the Tarjan algorithm. The measurements are done using a reinversion frequency of 60. During the tests the averages of transformations are computed. Solution times are analyzed in the next section.

The results are shown in Table 2. The first set of measurements is generated using threshold pivoting within the blocks of the block-triangular form, considering the column order given by the Tarjan algorithm. The second set of measurements is generated using threshold pivoting with the reordering of columns based on the column counts. The table also shows the reduction achieved in the number of transformations.

Table 2 Inversion statistics on average transformation numbers per inversion.

Problem name	Average number of transformations per inversion		Reduction
	Without reordering	Reordered	
25FV47	199757	130262	34.79 %
80BAU3B	386	291	24.40 %
BNL2	14782	8372	43.36 %
D2Q06C	1371770	739189	46.11 %
DEGEN3	237036	163321	31.10 %
FIT2D	4545	4033	11.26 %
FIT2P	215598	201894	6.36 %
GREENBEA	37640	15372	59.16 %
GROW22	196210	44865	77.13 %
MAROS-R7	211474	235210	-11.22 %
PILOT	11508000	10360700	9.97 %
PILOT87	47395100	36729000	22.50 %
QAP08	3859540	2387350	38.14 %
STOCFOR3	1457	1349	7.37 %
TRUSS	295365	270876	8.29 %
WOOD1P	30702	28133	8.37 %

The table clearly shows that reordering reduces the number of transformations required to compute the product form of the inverse in most cases. It can happen that the sequence of bases is different in these two cases because the numerical properties of the basis can produce numerically different reduced costs, thus the pricing strategy may choose different variables to enter the basis. This happened during the solution of MAROS-R7 in the reordered case. Bases with larger non-triangular parts have been obtained throughout the solution process.

The computation of the FTRAN and BTRAN operations become faster, too, because the resulting inverse usually has fewer non-zeros than it has in the general case. Reordering can be done very quickly because the column counts are directly available. It is important to note that the column reordering has other advantageous features. Most importantly, it provides a numerically more stable inverse, so reinversion frequency can be increased.

4.2 Investigating reinversion frequencies

In the literature the main problem with the PFI is that reiversion must be performed too frequently. Reinversion frequency of 30-60 is advised to be used. In this section we demonstrate the stability of our inversion process by increasing the reinversion frequency. This gives a substantial reduction in solution time if a problem cannot be triangularized well. Measurements have been made using both variants of the algorithm.

In Table 3 the test set with reinversion frequencies of 60, 120 and 300 and 1200 is presented without using the column ordering technique. In Table 4 the same measurements are shown with the reordering variant. The "Improvement" column shows comparison of the best solution time relative to the column of 60, which is said to be the maximal advised value. It is important to note that the column of 60 in Table 4 is already better (in most cases) than the same column in Table 3. This observation makes it clear that total improvement achieved by reordering combined with an increased reinversion frequency is more than the "Improvement". This result is shown in the "Total improvement" column.

The results clearly show that significantly more iterations can be done without reinverting than advised. Our method seems to be numerically stable. We can solve nearly all netlib problems with the current ("work in progress") version of PanOpt. Even PILOT and PILOT87, which are known to be numerically hard problems, can be solved with doing more than a hundred iterations in a row using the reordering variant. Remember, scaling is not used on the matrix, which proves that the numerical difficulties of the problems are handled properly by the inversion process and the FTRAN, BTRAN implementations.

Problem STOCFOR3 has a very good structure. Its non-triangular part and its average transformation count during inversion are relatively small in contrast to the problem size, thus the inversion process is fast. Unfortunately, as the sequence of ETMs representing the inverse gets longer, the BTRAN and FTRAN operations slow down and become less accurate. For such problems frequent reinversion is beneficial.

On the other hand, QAP08 and PILOT87 are extremely non-triangular, their transformation average is very high. Therefore, creating the inverse of the bases of these problems takes significant amount of time. Such problems benefit from more frequent reinversions as it can be seen in the "Improvement" columns.

It is also interesting to compare Tables 3 and 4. It shows that the reordering method generally decreases solution times. For PILOT using a reinversion frequency of 1200 solution cannot be obtained without column reordering because serious numerical problems occurred. On the other hand, with the reordered variant solution can be obtained for PILOT87 (1951.971 seconds) with reinverting only after 3000 iterations. Such results with the PFI have not been published yet.

5 Conclusions

In our work we have revisited the product form of the inverse for the simplex method. We have introduced a technique, which is appropriate to process the non-triangular part of the basis. The novelty of our approach is based on the block-triangular form. Our approach reorders the columns within each block based on the number of non-zeros in the columns. The threshold pivoting procedure is applied after reordering. In this way the resulting inverse is stable enough to be updated for hundreds or thousands of iterations and also solution times are reduced. We have implemented our method and presented a computational study to prove its efficiency.

We also have some further improvement ideas for the PFI. These ideas aim the further reduction of the number of transformations necessary to carry out inversions. Hopefully, stability will be further increased while the transformation count and solution time reduced. Any success will be reported in the future.

Table 3 Solution times with different reinversion frequencies (without reordering).

Problem name	Solution time (sec) using reinversion frequency				Best case	Improvement
	60	120	300	1200		
25FV47	15.581	12.667	18.735	48.670	12.667	18.70 %
80BAU3B	5.617	6.294	5.529	10.317	5.529	1.57 %
BNL2	9.617	12.365	14.685	63.884	9.617	0.00 %
D2Q06C	240.290	257.150	275.087	634.711	240.290	0.00 %
DEGEN3	26.811	27.299	33.402	67.905	26.811	0.00 %
FIT2D	24.774	24.619	25.271	28.135	24.619	0.63 %
FIT2P	114.350	132.001	191.432	680.007	114.350	0.00 %
GREENBEA	42.721	36.335	48.179	143.892	36.335	14.95 %
GROW22	1.067	0.947	1.024	2.178	0.947	11.25 %
MAROS-R7	28.116	23.214	24.114	37.175	23.214	17.43 %
PILOT	541.016	353.776	579.610	Error	353.776	34.61 %
PILOT87	3225.252	1988.138	1225.894	1308.544	1225.894	61.99 %
QAP08	68.090	40.026	44.149	84.112	40.026	41.22 %
STOCFOR3	58.828	63.529	87.504	217.524	58.828	0.00 %
TRUSS	35.959	23.461	31.171	60.112	23.461	34.76 %
WOOD1P	0.915	0.963	0.927	1.071	0.915	0.00 %

Table 4 Solution times with different reinversion frequencies (with reordering).

Problem name	Solution time (sec) using reinversion frequency				Best case	Improvement	Total improvement
	60	120	300	1200			
25FV47	11.604	10.957	13.553	46.733	10.957	5.58 %	29.68 %
80BAU3B	5.588	5.172	5.564	8.057	5.172	7.44 %	7.92 %
BNL2	10.576	12.386	13.770	51.131	10.576	0.00 %	-9.97 %
D2Q06C	178.310	152.192	217.553	557.962	152.192	14.65 %	36.66 %
DEGEN3	23.746	20.659	25.053	74.886	20.659	13.00 %	22.95 %
FIT2D	24.477	24.788	25.106	28.156	24.477	0.00 %	1.20 %
FIT2P	126.502	117.152	188.609	736.908	117.152	7.39 %	-2.45 %
GREENBEA	26.952	30.919	46.467	116.071	26.952	0.00 %	36.91 %
GROW22	0.858	0.928	0.948	2.150	0.858	0.00 %	19.59 %
MAROS-R7	28.019	23.887	24.230	37.412	23.887	14.75 %	15.04 %
PILOT	517.893	307.454	277.950	402.567	277.950	46.33 %	48.62 %
PILOT87	2884.267	1517.950	1089.021	1205.336	1089.021	62.24 %	66.23 %
QAP08	60.738	34.163	28.481	68.982	28.481	53.11 %	58.17 %
STOCFOR3	58.952	63.318	87.061	218.299	58.952	0.00 %	-0.21 %
TRUSS	25.518	24.662	28.309	59.582	24.662	3.35 %	31.42 %
WOOD1P	0.815	0.882	0.873	1.332	0.815	0.00 %	10.93 %

Acknowledgments We would like to thank our colleagues for their work in the implementation of other modules of Pannon Optimizer. This publication/research has been supported by the TÁMOP-4.2.2/B-10/1-2010-0025 project.

References

1 COIN-OR Linear Optimization Solver. `http://www.coin-or.org/projects/Clp.xml`.

2 G.B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In T.C. Koopmans, editor, *Activity analysis of production and allocation*, pages 339–347. Wiley, New York, 1951.

3 George B. Dantzig and Wm. Orchard-Hays. The product form for the inverse in the simplex method. *Mathematical Tables and Other Aids to Computation*, 8(46):64–67, 1954.

4 I. S. Duff and J. K. Reid. An implementation of Tarjan's algorithm for the block triangularization of a matrix. *ACM Transactions on Mathematical Software (TOMS)*, 4(2):137–147, 1978.

5 A. M. Erisman, R. G. Grimes, J. G. Lewis, and Jr. W. G. Poole. A structurally stable modification of Hellerman-Rarick's P4 algorithm for reordering unsymmetric sparse matrices. *SIAM Journal on Numerical Analysis*, 22(2):369–385, 1985.

6 E. Hellerman and D. Rarick. The partitioned preassigned pivot procedure (P4). In *Sparse matrices and their applications*, pages 67–76. Plenum Press, 1972.

7 Harry M. Markowitz. The elimination form of the inverse and its application to linear programming. *Management Science*, 3(3):255–269, 1957.

8 I. Maros. *Computational Techniques of the Simplex Method.* Kluwer Academic Publishers, Norwell, Massachusetts, 2003.

9 R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

The design of transportation networks: a multi objective model combining equity, efficiency and efficacy

Maria Barbati[1]

1 Department of Management Engineering, University of Naples Federico II
P.le Tecchio, 80, Italy
maria.barbati@unina.it

Abstract

A network design problem consists in locating facilities (nodes and arcs) that enable the transfer of flows (passengers and/or goods) from given origin-destination pairs. The topic can have several applications within transportation and logistics contexts. In this work we propose a multi-objective model in which balancing or equity aspects, i.e. measures of the distribution of distances of users from the path, are considered. These kinds of models can be used when there is the need to balance risks or benefits among all the potential users deriving from the location of the path to be designed. The application of the proposed model to a benchmark problem used in the literature to test these kinds of models, shows that it is able to find solutions characterized by significant level of equity but also of efficiency and efficacy.

1998 ACM Subject Classification G.1.6. Optimization

Keywords and phrases Graphs/Network, Transportation, Multi criteria Decision Analysis

Digital Object Identifier 10.4230/OASIcs.SCOR.2012.75

1 Introduction

A location problem consists in the positioning of a set of facilities within a given space. The decision is made on the basis of an objective function, which can concern the minimization of costs or the maximization of benefits.
If the facilities have an extension such that the representation through points is ineffective we deal with a network design problem. For example communication systems, public transport and energy distribution require appropriate networks for their representation.
A network $G(N, A)$ consists of a set of nodes, $N = 1, ..., n$, and a set of arcs $A = ((i, j) : i, j \in N)$. At each arc there is usually associated a cost while at each node can be associated a demand service.
A network design problem requires the definition of a subset of arcs to be inserted in a solution in order to optimize an objective function subject to a set of constraints. In particular topological constraints which are often included in the problem formulation require that the solution presents given topological characteristics (path, tree, cycle, network).
The objective function can be related to an efficiency measure (i.e. total network cost) or to an efficacy measure concerning demand satisfaction aspects (i.e. cost, accessibility)[6], [8]. Many models consider the simultaneous presence of more objectives (multicriteria network design problem [1], [2], [3]).
In addition to the mentioned criteria, in many applications the need to obtain solutions related to the concept of equity occurs. This means that in the network design one could

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn; pp. 75–83
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

search for solutions in which some parameters (cost and/or benefits) are distributed as evenly as possible among the potential users.
The importance of taking into account such aspects derives from various considerations [7]. In general if users perceive a substantial equity in the treatment of the fruition of a service, they are more satisfied. In addition when facilities are considered "undesirable", an equitable distribution of the risk and/or disadvantage due to their locations can reduce the conflicts among users and can help in accepting possible solutions.
In this paper we propose a path location problem in which balancing aspects are explicitly considered. The remainder of the paper is organized as follows. In the following section we illustrate the main path location models proposed in the literature. Then we propose a formulation of a new model. Computational experiments are then performed, in order to analyze solutions provided by the proposed model; finally, some conclusions and directions for further researches are drawn.

2 Path Location Problem

Let $G = (N, A)$ a network of potential arcs which can be included in a solution and $(O, D) \in N$ a pair of nodes. A path location problem consists of selecting a subset of arcs, according to some criteria and defining a path from the origin O to the destination D.
In the literature different path location models have been proposed. The Maximum Coverage Shortest Path Problem (MCSP) [4] was formulated as a problem with two objectives. It was assumed that there is a demand for each node and that this demand is covered by a path if the path passes to some node located within a given distance (threshold).
The first objective is to identify the shortest (or minimum cost) path while the second objective is to maximize the total demand covered by the path. These two objectives are usually conflicting as in general when we increase the length of the path, the covered demand also increases.
The Maximum Population Shortest Path Problem (MPSP) [5] is a special case of the MCSP problem, where the threshold is zero (i.e. the demand is satisfied at a node if that node belongs to the path). Variants of MCSP can include constraints on the distances between any node not belonging to the path and the path itself (mandatory closeness constraints).
The Median Shortest Path Problem (MSPP) [5] is another bi-criteria path location model where the second objective is oriented to maximize the "accessibility" to the path. This can be measured as the sum of the distances from any node to the closest node of the path. In practice the objective aims at minimizing the average cost to reach the path.
We introduce the following notation:
N_i = set of nodes j such that the arc (i, j) exists
M_j =set of nodes i such that the arc (i, j) exists
P_i = set of nodes j such that the path from i to j exists
w_i = demand associated to the node i
d_{ij}= distance between node i and j
T_{ij} = the length of the shortest path connecting node i to node j
Q = a non empty subset of N
$|Q|$ = the cardinality of subset Q

$$X_{ij} = \begin{cases} 1 & \text{if a direct arc between the node } i \text{ and the node } j \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{ij} = \begin{cases} 1 & \text{if node } i \text{ is assigned to node } j \\ 0 & \text{otherwise} \end{cases}$$

The MSPP can be formulated as:

$$\min Z = (Z_1, Z_2) = (\sum_i \sum_j d_{ij} X_{ij}, \sum_i \sum_j w_i T_{ij} Y_{ij}) \quad (1)$$

$$\text{s.t.} \sum_{j \in N_O} X_{Oj} = 1 \quad (2)$$

$$\sum_{i \in N_D} X_{iD} = 1 \quad (3)$$

$$\sum_{i \in M_j} X_{ij} - \sum_{k \in N_j} X_{jk} = 0 \qquad \forall j \in N, j \neq O, j \neq D \quad (4)$$

$$\sum_{j \in N_i} X_{ij} + \sum_{j \in P_i} Y_{ij} = 1 \qquad \forall i \in N, i \neq O, i \neq D \quad (5)$$

$$Y_{ij} - \sum_{i \in M_j} X_{ij} \leq 0 \qquad \forall (i,j) \in A \quad (6)$$

$$\sum_{i \in Q} \sum_{j \in Q} X_{ij} \leq |Q| - 1 \qquad \forall Q \subseteq N, |Q| \geq 2 \quad (7)$$

$$X_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \quad (8)$$

$$Y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \quad (9)$$

The objective function (1) is composed by two objectives. The first one (Z_1) measures the length (cost) of the path, while the function Z_2 measures the accessibility for each node. The constraints (2) and (3) assure, respectively, that the origin node and the destination node are on the median shortest path. The constraints set (4) states that demand of arcs entering and leaving a node is equal. The set of constraints (5) requires that each node is either in the path or is assigned to a node that is on that path. The set of constraints (6) prevents a node i, which is not on the path, to be assigned to a node j that does not belong to the path. The set of constraints (7) avoids the possible presence of cycles. The last constraints sets (8) and (9) ensure that the variables X_{ij} and Y_{ij} are binary.
Moreover in the Equity Constrained Shortest Path Problem, introduced in [9], [10] a path is considered feasible if the sum of the differences in cost between all pairs of nodes is less than a certain threshold value, called equity parameter. The introduction of equity for network design is also analyzed in [11].

3 The proposed model

We propose a version of the MSPP described above in order to include balancing aspects in the solution to be found. To this aim we introduce the parameter $r_k(i,j)$ defined as the risk or benefit perceived at node k due to the presence of the arc (i,j). We can then consider the total perceived cost or benefit at node k as the sum of the cost $r_k(i,j)$ due to any arc (i,j) belonging to a path P. We can assume that $r_k(i,j)$ depends on the distance from k to the closest node between i and j.
This way a balancing constraint can be defined as:

$$w_k \cdot (\sum_{(i,j) \in P} r_k(i,j)) - w_h \cdot (\sum_{(i,j) \in P} r_h(i,j)) \leq \mu \qquad \forall k, h \in N \quad (10)$$

where μ is an equity parameter representing the maximum cost or benefit difference between any pair of nodes k and h. Adding this constraints to the model (1) – (9) we could obtain a formulation that combines efficiency (the minimization of path length), efficacy

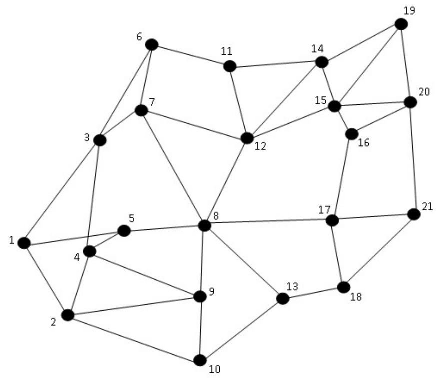


Figure 1 The used test problem.

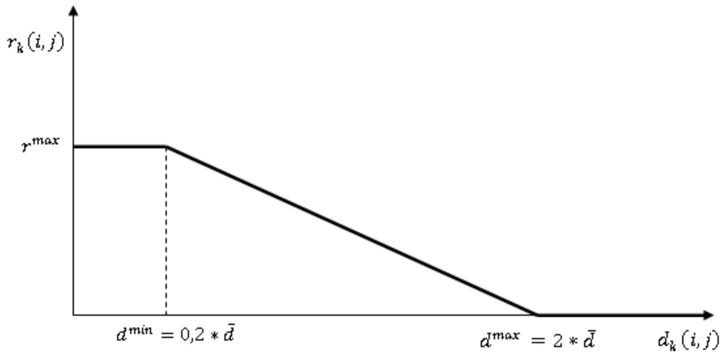


Figure 2 Expression of $r_k(i,j)$.

(the maximization of the accessibility), and the minimization of the disequity, i.e. better distribution of the cost or the benefit among the users.

4 Computational experiments

For testing our model we used a benchmark network introduced in [5] and represented by the graph in Figure 1 with 21 nodes and 39 arcs. Each node is associated a demand value and each arc is associated a length; finally, the distance $d(i,j)$ is associated to each pair of nodes (i,j).

Indicating with $d_k(i,j) = \min(d(i,k), d(j,k))$, in order to set the values of $r_k(i,j)$, we assumed that $r_k(i,j)$ is equal to a maximum value r^{max} if $d_k(i,j) < 0.2 * \overline{d}$ where $\overline{d}$ is the average of the distances; if $0.2 * \overline{d} < d_k(i,j) < 0.2 * d$ then $r_k(i,j)$ decreases linearly from r^{max} reaching 0 for $d_k(i,j) > 2 * \overline{d}$ (see Figure 2).

The arcs are divided into three categories corresponding to the values:

- $r^{max} = 3$;
- $r^{max} = 2$;
- $r^{max} = 1$.

A higher value of r^{max} indicates a bigger risk associated with that arc. In Figure 3 we report a representation of the three categories of arcs, identifying the different categories with different dotted lines.

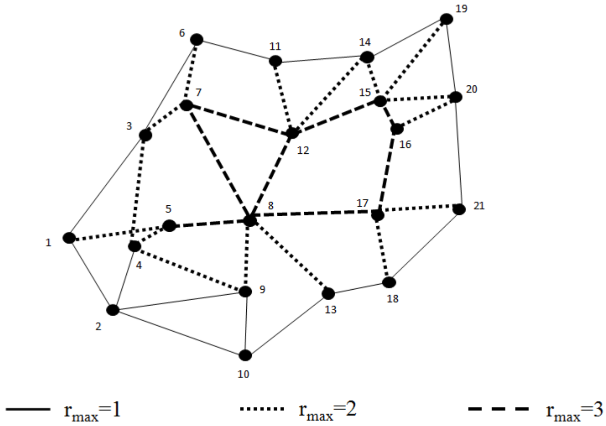


Figure 3 Arc categories.

In order to solve the model we consider the single-objective version of the MSPP model obtained by introducing a linear combination of the objective function $Z = (Z_1, Z_2)$ equal to:

$$Z^{'} = \lambda \cdot Z_1 + (1 - \lambda) \cdot Z_2 \tag{11}$$

with λ included between 0 and 1. This way the optimal solution of the model $(1) - (9)$, assuming Z is simple objective function with a fixed value of λ, is a Pareto solution for the model $(1) - (9)$.

We solved the MSPP model and the proposed variant with the addition of constraints (10), using the software CPLEX 12.0. and throughout all the testing, we used a Pentium IV with 2.40 GHz and 4.00 GB of RAM running. The values of the equity parameter μ were fixed in order to assure that equity constraint permits to obtain different solutions from those provided by the model MSPP. For the considered instance we found the appropriate value of μ by iteratively solving the model in order to find the minimum value of μ $(1.8 * 10^{-6})$ in such a way that the model provides at least one feasible solution. Starting from this minimum value, μ is then increased with a step of $0.1 * 10^{-6}$ until constraints (10) are not active.

The computational times are very low; for all the analyzed instances we found the optimal solution in less than one minute.

In Figure 4 we report the path obtained for each value of the weight λ, from 0 to 1 with a step equal to 0.1. When $\lambda = 0.0$ as the objective function aims at minimizing the accessibility cost, the path visits each node of the graph (solution (a)). As λ increases, the number of visited nodes decreases until $\lambda \geq 0.5$ and the shortest path is obtained (solution (e)).

In Figure 5 we show the solutions provided by the proposed model with different values of λ assuming $\mu = 2.2 * 10^{-6}$. The obtained solutions do not correspond to those shown in Figure 4. In particular with $\lambda = 0.0$ the feasible solution which minimizes the accessibility cost (solution (f)) visits 10 of 21 nodes, with most of the arcs characterized by lower risk values. With values $\lambda = 0.1, 0.2, 0.3$ the solution is quite similar to those provided by the MSPP but with a selection of arcs with lower risk values. Finally with $\lambda \geq 0.4$ the solution is similar to the one founded by the MSPP with $\lambda \geq 0.5$.

In Figure 6 we report the value of $Z^{'}$ calculated through (11) for the two models varying both the parameters μ and λ.

We can highlight that the optimal solution of the MSPP for a given value of λ represents a

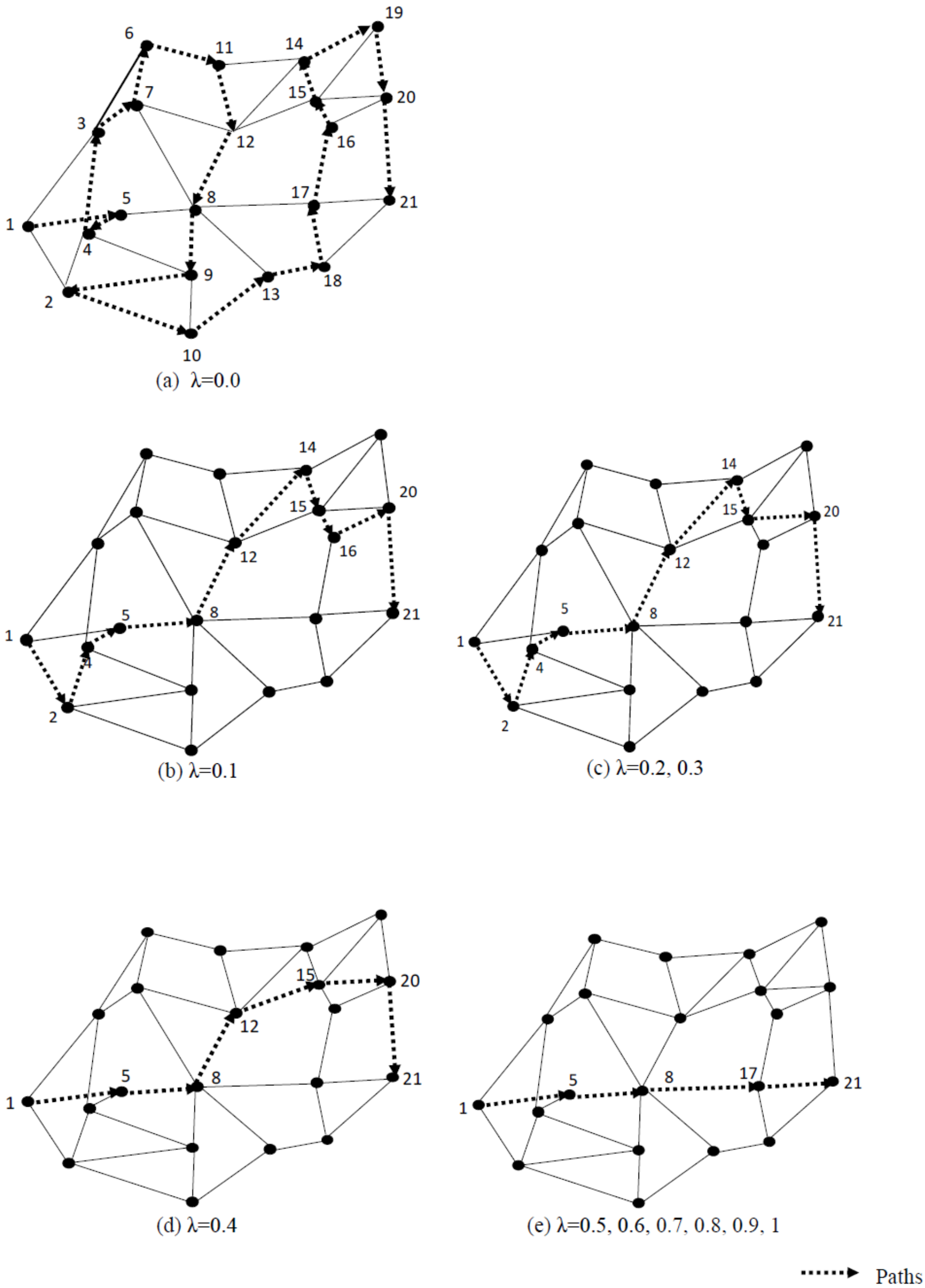


Figure 4 Solution for the MSPP by varying λ $(0 \leq \lambda \leq 1)$ with step 0.1.

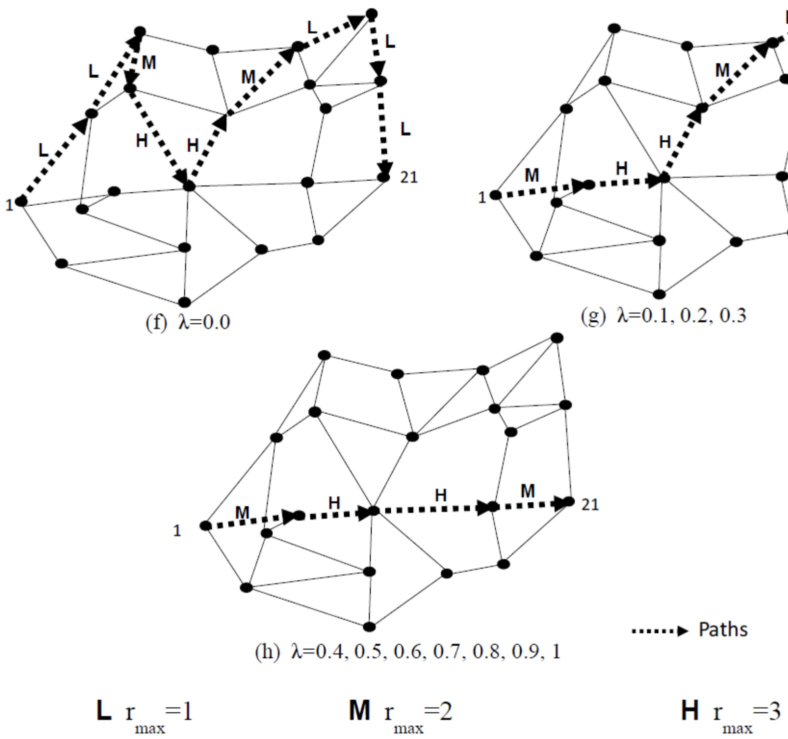


Figure 5 Solutions for the MSPP adding equity constraints with $\mu = 2.2 * 10^{-6}$, by varying λ ($0 \leq \lambda \leq 1$) with step 0.1.

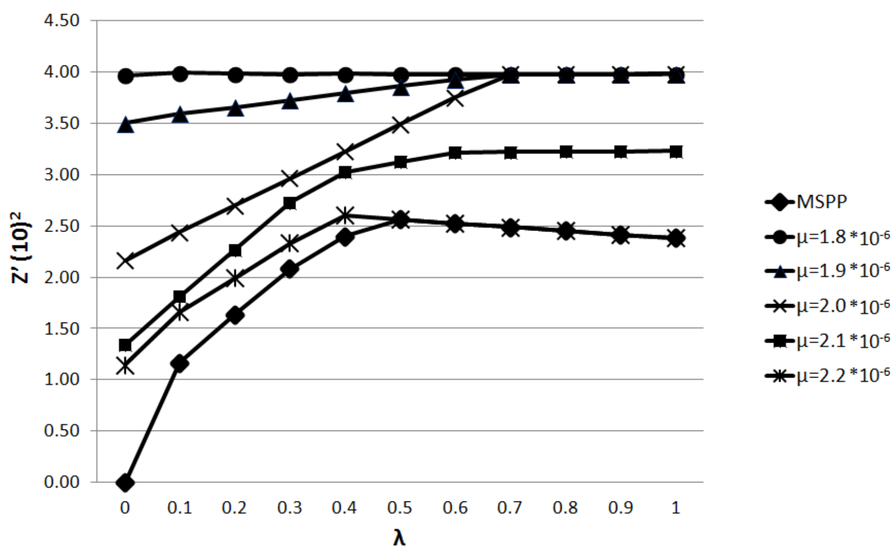


Figure 6 Values of $Z^{'}$ by varying μ and λ.

lower bound for Z when the equity constraint is introduced. When μ decreases we obtain solutions with significant difference from the lower bound due to the fact that the set of equity constraints is more active and then its presence reduces the set of feasible solutions. For example analyzing the obtained solution for $\lambda = 0.5$ for the lowest value of μ, and so for the highest level of equity, the solution is characterized by a higher value of $Z^{'}$. Increasing μ, and so decreasing the level of the equity, also the level of $Z^{'}$ decrease; so, the solution for intermediate values of μ are characterized by both level of equity and $Z^{'}$ (representing of the other two objectives), as wished. In addition, decrementing the level of equity the solution becomes more similar to those obtained by the MSPP model, until when $\mu \geq 2.3$ for which the solutions of the proposed model correspond with ones provided by the MSPP model for all the values of λ; in practice the constraint is not more effective.

5 Conclusion

In this work we analyzed the path location problems, taking into account the three main models in the literature: the Median Shortest Path Problem (MSPP), the Maximum Covering Shortest Path Problem (MCSPP) and the Equity Constrained Shortest Path problem. From the analysis of the literature we found the opportunity to include equity aspects in this context. For this reason we proposed a variant of the MSPP introducing balancing constraints. We formulated the model and we tested its effectiveness through experiments on a test problem. Comparing the solutions, it became evident the effect of the insertion of the equity constraints on the resulting paths; indeed we found paths with a higher level of equity and also with good values of the other two objectives. As further research we want to seek alternative methods of calculating the perceived risk from each node, in order to adapt the model to describe different applications.

Acknowledgements I would like to thank Riccardo Palladino for the work completed during his degree thesis from which this work originates and my supervisor Giuseppe Bruno for helping me to develop this work and also this paper.

References

1 B. Boffey *Multiobjective Routing Problems*. Top, vol.3, pp. 167-203, 1995.
2 J.R. Current. *Multiple objectives in transportation network design and routing*. European Journal of Operational Research vol.65, pp. 1-3, 1993.
3 J. Current, M.Marsh. *Multiobjective transportation network design and routing problems:Taxonomy and annotation*. European Journal of Operational Research vol.65, pp. 4-19, 1993.
4 J.R. Current , C.S. Revelle, J.L. Cohon. *The maximum covering/shortest path problem: a multiobjective network design and routing formulation*. European Journal of Operational Research, vol. 21, pp. 189-199, 1985.
5 J.R. Current , C.S. Revelle, J.L. Cohon. *The median shortest path problem: a multiobjective approach to analyze cost vs. accessibility in the design of transportation networks*. Transportation Science, vol. 21, pp. 188-197, 1987.
6 M. Ehrgott, X. Gandibleux *A survey and annotated bibliography of multiobjective combinatorial optimization*. Operations Research Spektrum, vol. 22, pp. 425-460, 2000.
7 H.A. Eiselt, G. Laporte *Objectives in Location Problems. In: Facility Location: A Survey of Applications and Methods*, pp.151-180,Springer-Verlag, Berlin, 1995.
8 R.Z. Farahani, M. SteadieSeifi, N. Asgari. *Multiple criteria facility location problems: A survey*. Applied Mathematical Modelling, vol. 34, 1689-1709, 2008.

9 R. Gopalan, R. Batta, M.H. Karwan . *The Equity Constrained Shortest Path Problem*. Computers & Operations Research, vol.17, pp. 297-307, 1990.

10 R. Gopalan, K.S. Kolluri, R. Batta, M.H. Karwan. *Modeling equity of risk in the transportation of hazardous materials*. Operations Research, vol.38, pp. 961-973, 1990.

11 M.B. Mandell. *Modelling Effectiveness-Equity Trade-Offs in Public Service Delivery Systems*. Management Science, vol. 37, pp. 467-482, 1991.

Heuristics for the routing of trucks with double container loads*

Michela Lai[1], Massimo Di Francesco[2], and Paola Zuddas[3]

1 Department of Mathematics and Computer Science, University of Cagliari
Via Ospedale 72, 09124, Cagliari, Italy
mlai@unica.it
2 Department of Land Engineering, University of Cagliari
Via Marengo 2, 09123, Cagliari, Italy
mdifrance@unica.it
3 Department of Mathematics and Computer Science, University of Cagliari
Via Ospedale 72, 09124, Cagliari, Italy
zuddas@unica.it

Abstract

This research addresses a problem motivated by a real case study. A carrier must plan the routes of trucks in order to serve importers and exporters. What is original in this vehicle routing problem is the impossibility to separate trucks and containers during customer service and the opportunity to carry up to two containers per truck. Customers may demand more than one container and may be visited more than once. Moreover, according to the carrier's policy, importers must be served before exporters. In order to address this Vehicle Routing Problem with backhaul and splits, a linear integer programming model is proposed. This research aims to show to what extent an exact algorithm of a state of the art solver can be used to solve this model. Moreover, since some instances are too difficult to solve for the exact algorithm, a number of heuristics is proposed and compared to this algorithm. Finally, the heuristics are compared to the real decisions of the carrier who has motivated this problem.

1998 ACM Subject Classification H.4.2 Types of Systems Logistics

Keywords and phrases Split Vehicle Routing Problem, Backhaul, Drayage, Container transportation, Heuristics

Digital Object Identifier 10.4230/OASIcs.SCOR.2012.84

1 Introduction

This paper addresses a vehicle routing problem, which is motivated by a real case study. A carrier is in charge of planning the distribution of container loads by trucks and containers based at a port. The carrier has a homogeneous fleet of trucks carrying up to two containers and the planning of routes must be performed within 10 minutes. Two classes of customers must be served: importers and exporters. The importers need to receive full container loads from the port and the exporters need to ship container loads to the port. Typically customers need to be served by more than one container and must be visited by more than one truck.

According to the carrier's policy, trucks and containers cannot be uncoupled during customer service, because truck drivers are required to check the right execution of operations. As a result, in this problem there are no pickups or deliveries of loaded and empty containers:

* This work was partially supported by Grendi Trasporti Marittimi.

during customer service containers are filled or emptied and moved away by the same trucks used for bringing containers to customers.

Moreover, since the container loads of exporters are typically not ready before the afternoon, the carrier policy is to serve importers before exporters. As a result, empty containers leaving from importers can be moved to exporters, where they are filled and shipped to the port.

This problem belongs to the class of Vehicle Routing Problem (VRP) with backhauls, because deliveries must be performed before pickups [1]. However, in classical VRP with backhaul each customer must be visited only once, whereas in our problem multiple visits at each customer are allowed. Therefore, although several solution methods exist in VRP with backhauls [2], [3], [4], [5], they may be suboptimal because splits are not considered.

There are also some similarities to drayage problems, which consists of picking up and delivering of full containers. Typically they are separated from trucks during customer service [6], whereas this is not possible in this problem. The closest problem setting was probably faced by Imai et al. [7], who studied the optimal assignment of own and chartered trucks to a set of delivery and pickup pairs. As in our setting, tractors and containers cannot be uncoupled, but the capacity of trucks is limited to one container only. Homogeneous fleets with one container trucks are also considered in [8] and [9].

The objective of this paper is to propose an optimization model accounting for the original characteristics of this problem. The model minimizes distribution costs, such that all customers are served as requested, truck capacity constraints hold and importers are served before exporters.

Since we are required to determine efficient solutions rapidly, this paper also aims to propose a number of heuristics for this problem. The common idea between these heuristics is to build an initial set of routes in which all customers are either importers or exporters. Next, these routes are merged according to different criteria, one for each heuristic.

The contributions of this paper are:

- to present a problem with some original characteristics, which have not been investigated in the rich VRP literature;
- to model the problem by a linear integer programming formulation;
- to propose and evaluate a number of heuristics with respect to an exact algorithm of a state-of-art solver.

The paper is organized as follows. In Section 2 the problem description is presented. The problem is modeled in Section 3. Solution methods are described in Section 4. The heuristics are tested in Section 5. Section 6 presents a summary of conclusions and describes future research perspectives in the field.

2 Problem description

Consider a fleet of trucks and containers based at a port. Trucks carry up to two containers and serve two types of customer requests: the delivery of container loads from the port to importers and the shipment of container loads from exporters to the same port. Typically customers need to ship or receive more than one container load. Therefore, usually each customer must be served by multiple containers and must be visited more than once.

A relevant characteristic of this problem is the impossibility to separate trucks and containers during customer service. As a result, when importers receive container loads by trucks, containers are emptied and moved away by the same trucks used for providing container loads. Similarly, when exporters are served by empty containers, containers are filled and moved away by the same trucks used for providing empty containers.

According to the carrier's policy, importers must be served before exporters. As a result, routes may consist in the shipment of container loads from the port to importers, the direct allocation of empty containers from importers to exporters and the final shipment of container loads from exporters to the port. Therefore, trucks can serve in a route up to four customers (two importers and two exporters). Every pair of containers can be shipped in a truck. All containers leaving from importers can be used to serve exporters, no incompability occurs between customers and trucks, which can serve almost any customer, and there are no priorities among importers and among exporters.

It is worth noting that the number of container loads to be picked up and delivered is generally different. When the number of container loads delivered to importers is larger than the number of container loads shipped by exporters, several empty containers must be moved back to the port. When the number of container loads delivered to importers is lower than the number of container loads shipped by exporters, several empty containers must be put on trucks leaving from the port, in order to serve all customers.

The movement of trucks generate routing costs. In this problem, all trucks lead to the same routing costs per unitary distance. Moreover, handling costs are paid to put containers on trucks at the port. The objective is to determine the routes of trucks in order to minimize routing and handling costs, such that customers are served as requested, truck capacity constraints hold and importers are served before exporters.

3 Modeling

We consider a port p, a set I of importers, a set E of exporters, and a set K of different trucks, whose transportation capacity is 2 containers. An integer demand of $d_i \geq 0$ containers is associated with each customer $i \in I \cup E$. It represents the number of loaded containers requested to serve import customer $i \in I$ and it is also equal to the number of empty containers returned by this customer. When $i \in E$, d_i represents the number of empty containers requested to serve export customer $i \in E$ and it is also equal to the number of loaded containers shipped by this customer.

Consider a direct graph $G = (N, A)$, where $N = \{p \cup I \cup E\}$ and the set of arcs A includes all allowed ways to move trucks:

- from the port to any importer and any exporter;
- from an importer to the port, any other importer and any exporter;
- from an exporter to the port and any other exporter.

More formally, the set A is defined as $A = A_1 \cup A_2$, where

$$A_1 = \{(i, j) | i \in I \cup p, j \in N, i \neq j\}$$
$$A_2 = \{(i, j) | i \in E, j \in E \cup p, i \neq j\}$$

The operation cost c_{ij} for any truck traversing arc $(i, j) \in A$ is supposed to be nonnegative. Let h_{pj} be the nonnegative handling cost of a container put on and picked from any truck at the port p to serve node $j \in N$.

The following decision variables are defined:

- x^k_{ij}: Routing selection variable equal to 1 if arc $(i, j) \in A$ is traversed by truck $k \in K$, and 0 otherwise;
- y^k_{ij}: Integer variable representing the number of loaded containers moved along arc $(i, j) \in A$ by truck $k \in K$;

- z_{ij}^k: Integer variable representing the number of empty containers moved along arc $(i,j) \in A$ by truck $k \in K$.

The problem can be formulated as follows:

$$min \sum_{k \in K} \left[\sum_{(i,j) \in A} c_{ij} x_{ij}^k + \sum_{j \in N} h_{pj}(y_{pj}^k + z_{pj}^k) \right] \quad (1)$$

$s.t.$

$$\sum_{k \in K} \sum_{l \in N} y_{il}^k = \sum_{k \in K} \sum_{j \in p \cup I} y_{ji}^k - d_i \quad \forall i \in I \quad (2)$$

$$\sum_{k \in K} \sum_{l \in N} z_{il}^k = \sum_{k \in K} \sum_{j \in p \cup I} z_{ji}^k + d_i \quad \forall i \in I \quad (3)$$

$$\sum_{l \in N} y_{il}^k \leq \sum_{j \in p \cup I} y_{ji}^k \quad \forall i \in I, \forall k \in K \quad (4)$$

$$\sum_{l \in N} z_{il}^k \geq \sum_{j \in p \cup I} z_{ji}^k \quad \forall i \in I, \forall k \in K \quad (5)$$

$$\sum_{k \in K} \sum_{l \in p \cup E} y_{il}^k = \sum_{k \in K} \sum_{j \in N} y_{ji}^k + d_i \quad \forall i \in E \quad (6)$$

$$\sum_{k \in K} \sum_{l \in p \cup E} z_{il}^k = \sum_{k \in K} \sum_{j \in N} z_{ji}^k - d_i \quad \forall i \in E \quad (7)$$

$$\sum_{l \in p \cup E} y_{il}^k \geq \sum_{j \in N} y_{ji}^k \quad \forall i \in E, \forall k \in K \quad (8)$$

$$\sum_{l \in p \cup E} z_{il}^k \leq \sum_{j \in N} z_{ji}^k \quad \forall i \in E, \forall k \in K \quad (9)$$

$$y_{ij}^k + z_{ij}^k \leq 2 x_{ij}^k \quad \forall (i,j) \in A, \forall k \in K \quad (10)$$

$$\sum_{j \in N} x_{ji}^k - \sum_{l \in N} x_{il}^k = 0 \quad \forall i \in N, \forall k \in K \quad (11)$$

$$\sum_{j \in N} x_{pj}^k \leq 1 \quad \forall k \in K \quad (12)$$

$$\sum_{k \in K} \sum_{i \in I \cup E} z_{ip}^k - \sum_{k \in K} \sum_{i \in I \cup E} z_{pi}^k = \sum_{i \in I} d_i - \sum_{i \in E} d_i \quad (13)$$

$$x_{ij}^k \in \{0,1\} \quad \forall (i,j) \in A, \forall k \in K \quad (14)$$

$$y_{ij}^k \in \{0,1,2\} \quad \forall (i,j) \in A, \forall k \in K \quad (15)$$

$$z_{ij}^k \in \{0,1,2\} \quad \forall (i,j) \in A, \forall k \in K \quad (16)$$

Container handling and truck routing costs are minimized in the objective function (1).

Constraints from (2) to (5) concern the service to importers. Constraints (2) and (3) are the flow conservation constraints of loaded and empty containers respectively at each importer node. Constraints (4) and (5) check the number of loaded and empty containers in each truck entering and leaving from importers: when a truck leaves from each importer, the number of loaded containers cannot increase and the number of empty containers cannot be reduced.

Constraints from (6) to (9) concern the service to exporters. Constraints (6) and (7) are the flow conservation constraints of loaded and empty containers, respectively, for each exporter node. Constraints (8) and (9) control the number of loaded and empty containers in each truck entering and leaving from exporters: when a truck leaves from each exporter, the number of loaded containers cannot be reduced and the number of empty containers cannot be increased.

Constraint (10) imposes that the number of containers moved by each truck is not larger than the transportation capacity. Constraints (11) are the flow conservation constraints for trucks at each node. Constraint (12) guarantees that trucks are not used more than once. Constraint (13) represents the flow conservation of empty containers at port p.

Finally, constraints (14), (15) and (16) define the domain of decision variables.

4 Solution methods

Several solution methods can be adopted to solve the previous problem. Generally speaking, they can be divided into exact and heuristic methods. In this paper we illustrate to what extent a well-known exact algorithm can be used to face this problem. This analysis is performed by the solver Cplex, which solves integer programming models by a branch-and-cut algorithm.

However, most Vehicle Routing Problems are *NP-hard* and, also in our problem, there is little hope of finding exact solution procedures for large problem instances [11]. Therefore, in this section we propose a number of heuristics, which can be used to tackle the problem at hand.

All proposed heuristics are composed of two phases. The first phase, which is the same for all heuristics, determines an initial solution, in which all routes serve either importers or exporters. In the second phase, each heuristic implements a different rule to merge the routes determined in the first phase. Finally, the best heuristic in terms of objective function is selected.

In the first phase, we face two vehicle routing problems with splits: the first has importers only, whilst the second has exporters only. However, Split Vehicle Routing Problems are also known to be difficult. Therefore, since an efficient metaheuristic for this class of problems has been proposed by Archetti et al. [10], their algorithm has been chosen to determine routes in the first phase.

The routes determined in the first phase are merged in the second phase according to different saving-based heuristics. Savings represent the routing costs achieved by merging two routes instead of leaving them separately. Given a route i with importers only and a route j with exporters only, the saving generated by their merging is computed as $s_{ij} = c(i) + c(j) - c(ij)$, in which $c(i)$ and $c(j)$ are the respective costs of routes i and j, and $c(ij)$ is the cost of the merged route. Savings are saved in a matrix, in which the number of rows is equal to the number of routes serving importers in the first phase and the number of columns is equal to the number of routes serving exporters in the first phase.

In this paper, the order of visits between pairs of importers and pairs of exporters is not changed after the merging. To clarify, let us consider for instance two importers i_1 and i_2 and two exporters e_1, e_2. Assume that the routes determined in the first phase are p, i_1, i_2, p and p, e_1, e_2, p. If these routes are merged, the final route is p, i_1, i_2, e_1, e_2,p. Therefore, the possibility of visiting importer i_2 before importer i_1 and exporter e_2 before e_1 is not taken into account.

Some definitions are provided for the sake of clarity in the presentation of the heuristics:

Row i represents the $i-th$ route of importers, as determined in the first phase;

Column j represents the $j-th$ route of exporters, as determined in the first phase;

Entry s_{ij} $>= 0$ is the saving generated by the merging of routes i and j. Only nonnegative savings are considered. When an entry s_{ij} takes value 0, the merging is not allowed. Whenever two routes i and j are merged by a heuristic, the related saving s_{ij} is set to 0;

m_i Number of columns (routes) that can be merged with the route represented by row i;

m_j Number of rows (routes) that can be merged with the route represented by column j;

$avrg_i$ is the average of all savings in row i;

$avrg_j$ is the average of all savings in column j.

We propose eight heuristics, whose solution is denoted by $s_{0,\ldots,7}$:

Heuristic 0 (H0) This heuristic does nothing and returns routes as determined in the first phase:

$Step_0$ $s_0 = \emptyset$.

$Step_1$ For each row i, insert i into s_0.

$Step_2$ For each column j, insert j into s_0.

Heuristic 1 (H1) This heuristic determines the maximum saving for each route of importers and selects the best routes serving exporters:

$Step_0$ $s_1 = \emptyset$.

$Step_1$ For each row i, select the largest s_{ij}. Merge routes i and j, if any, and insert the new route into s_1.

$Step_2$ For each row i not involved in any merging, insert i into s_1.

$Step_3$ For each column j not involved in any merging, insert j into s_1.

Heuristic 2 (H2) This heuristic determines the maximum saving for each route of exporters and selects the best routes serving importers:

$Step_0$ $s_2 = \emptyset$.

$Step_1$ For each column j, select the largest s_{ij}. Merge routes i and j, if any, and insert the new route into s_2.

$Step_2$ For each row i not involved in any merging, insert i into s_2.

$Step_3$ For each column j not involved in any merging, insert j into s_2.

Heuristic 3 (H3) This heuristic gives priority to routes of importers that can be merged with a low number of other routes.

$Step_0$ $s_3 = \emptyset$.

$Step_1$ Search for row i with the lowest value of m_i. If any, go to $Step_2$, otherwise go to $Step_3$.

$Step_2$ Select the largest s_{ij} for i, merge routes i and j and insert the new route into s_3. Go to $Step_1$.

$Step_3$ For each row i not involved in any merging, insert i into s_3.

$Step_4$ For each column j not involved in any merging, insert j into s_3.

Heuristic 4 (H4) This heuristic gives priority to routes of exporters that can be merged with a low number of other routes.

$Step_0$ $s_4 = \emptyset$.

$Step_1$ Search for column j with the lowest value of m_j. If any, go to $Step_2$, otherwise go to $Step_3$.

$Step_2$ Select the largest s_{ij} for j, merge routes i and j and insert the new route into s_4. Go to $Step_1$.

$Step_3$ For each row i not involved in any merging, insert i into s_4.

$Step_4$ For each column j not involved in any merging, insert j into s_4.

Heuristic 5 (H5) This heuristic gives priority to routes of both importers and exporters, that can be merged with a low number of other routes:

$Step_0$ $s_5 = \emptyset$.

$Step_1$ Search for row i with the lowest value of m_i and the column j with the lowest value of m_j. If $m_i <= m_j$, go to $Step_2$, otherwise go to $Step_3$. If no routes can be merged, go to $Step_4$.

$Step_2$ Select the largest s_{ij} for i, merge routes i and j and insert into s_5. Go to $Step_1$.

$Step_3$ Select the largest s_{ij} for j, merge routes i and j and insert into s_5. Go to $Step_1$.

$Step_4$ For each row i not involved in any merging, insert i into s_5.

$Step_5$ For each column j not involved in any merging, insert j into s_5.

Heuristic 6 (H6) This heuristic differs from the previous one in the selection of savings: we choose the closest saving to the average of all available savings, instead of the largest one:

$Step_0$ $s_6 = \emptyset$.

$Step_1$ Search for row i with the lowest value of m_i and the column j with the lowest value of m_j. If $m_i <= m_j$, go to $Step_2$, otherwise go to $Step_3$. If no routes can be merged, go to $Step_4$.

$Step_2$ Select the closest s_{ij} to $avrg_i$, merge routes i and j and insert into s_6. Go to $Step_1$.

$Step_3$ Select the closest s_{ij} to $avrg_j$, merge routes i and j and insert into s_6. Go to $Step_1$.

$Step_4$ For each row i not involved in any merging, insert i into s_6.

$Step_5$ For each column j not involved in any merging, insert j into s_6.

Heuristic 7 (H7) This heuristic merges the routes with the largest saving in the matrix:

$Step_0$ $s_7 = \emptyset$.

$Step_1$ Select for the largest s_{ij} in the saving matrix. If any, go to $Step_2$, otherwise go to $Step_3$.

$Step_2$ Merge routes i and j and insert the new route into s_7. Go to $Step_1$.

$Step_3$ For each row i not involved in any merging, insert i into s_7.

$Step_4$ For each column j not involved in any merging, insert j into s_7.

After the execution of all heuristics, we select the best one in terms of objective function.

5 Experimentation

In this section we test the previous heuristics on artificial and real instances. The real instances are provided by a shipping company operating in the port of Genoa (Italy).

Tests are performed on both artificial and real instances. Five classes of artificial instances have been generated:

- 10 customers;
- 20 customers;
- 30 customers;
- 40 customers;
- 50 customers.

In each class the coordinates of nodes are fixed. The instances of a class differ in the number of importers and exporters. The heuristics are implemented in the programming language C++. Tests are performed by Cplex 12.2 running on a four-CPU server 2.67 GHz

64 GB RAM. Since a major requirement of this problem is to determine solutions in a few minutes, Cplex is set to stop after 10 minutes. Computational results are indicated in Table 2, in which the following notation is used:

- $|I|$: Number of importers;
- $|E|$: Number of exporters;
- $H0$, ..., $H7$: Objective function returned by Heuristic 0,..., Heuristic 7;
- *t(s)*: The total execution time (in seconds) to solve the related instance, i.e. it represents how long it takes to run the first phase plus the time spent to run all heuristics $H0$, ..., $H7$;
- *% Gap from CPLEX* : gap between the best heuristic and the best upper bound provided by CPLEX within 10 minutes;
- *Optimality gap*: Optimality gap between lower and upper bounds in Cplex after 10 minutes.

The string *n.s.* means that Cplex cannot provide a feasible solution within 10 minutes.

Table 2 shows that only one instance with 10 customers can be optimally solved. Cplex does not provide feasible solutions within 10 minutes for all instances with 40 and 50 customers, whereas all heuristics can solve these instances within 10 minutes. Generally speaking, the heuristic $H7$ is the most promising in terms of the objective function.

Real instances, which have about 40 customers, have no feasible solutions by Cplex within 10 minutes. In this case we compare the best heuristic to the carrier's decisions in terms of total travelled distances. Results on the real instances are shown in Table 1, in which the following notation is used:

- *Instances* The instance considered;
- $|I|$ Number of importers;
- $|E|$ Number of exporters;
- $|K|$ Number of trucks;
- *Carrier's decisions* The total travelled distance according to the carrier's decisions (km);
- *Decisions* The total travelled distance according to the best heuristic (km);
- *% Improvement*: gap between the best heuristic and the carrier's decisions;
- *Criterion* The heuristic(s) providing the best solution;

Table 1 The solutions of real instances.

Instances	$\|I\|$	$\|E\|$	$\|K\|$	Carrier's decisions	Best Heuristic		
					Decisions	% Improvement	Criterion
Instance 1	7	34	41	16503	**16196**	-1.90	7
Instance 2	10	28	31	13369	**11701**	-14.26	7
Instance 3	3	31	39	13702	**13602**	-0.74	1, 3, 7
Instance 4	6	34	36	13263	**12328**	-7.58	7
Instance 5	3	28	41	13180	**12869**	-2.42	1, 3, 7

Table 1 shows that in each instance the best heuristic always improves the carrier's decisions. The improvement seems to be particularly relevant when $|I|$ increases and becomes closer to $|E|$, due to the larger search space of feasible routes.

6 Conclusion

This paper has investigated a vehicle routing problem with some original characteristics, such as the opportunity to carry two containers per truck and the impossibility to separate trucks and containers during customer service. We have formulated an integer linear programming model for this problem. An exact algorithm was used to solve several artificial instances, but it was able to solve only instances with few customers. Several heuristics are proposed and tested on both artificial and real instances. According to our tests, the most promising heuristic in terms of objective function is $H7$, because high-quality routes are built from the beginning by the maximum saving.

The comparison with the carrier's decisions shows that the heuristics represent a promising instrument to improve its current decision-making process, because they yield significant savings in distances travelled by trucks.

Research is in progress to face problems with heterogeneous fleets of trucks, time windows and larger transportation capacities. New heuristics will be developed accounting for the specific characteristics of these problems.

Acknowledgements The authors want to thank professor Claudia Archetti, who made available her Tabu Search code for the Split Vehicle Routing Problem. The authors are also grateful to Antonio Musso, managing director of Grendi Trasporti Marittimi, who introduced to them the application that motivated this research.

References

1 P. Toth, D. Vigo. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, USA, 2002.
2 S. Anily. *The Vehicle Routing Problem with Delivery and Back-haul Options*. Naval Research Logistic Quarterly, 43, 415-434, 1996.
3 A. Mingozzi, S. Giorgi and R. Baldacci. *An Exact Method for the Vehicle Routing Problem with Backhauls*. Transportation Science, 33, 315-329, 1999.
4 P. Toth, D. Vigo. *An Exact Algorithm for the Vehicle Routing Problem with Backhauls*. Transportation Science, 31, 372-385, 1997.
5 P. Toth, D. Vigo. *A Heuristic Algorithm for the Symmetric and Asymmetric Vehicle Routing Problem with Backhauls*. European Journal of Operational Research, 113, 528-543, 1999.
6 R. Zhang, W.Y. Yun, I.K. Moon. *Modeling and optimization of a container drayage problem with resource constraints*. International Journal of Production Economics, 133, 351-359, 2011.
7 A. Imai, E. Nishimura, J. Current. *A Lagrangian relaxation-based heuristic for the vehicle routing with full container load*. European Journal of Operational Research, 176, 87-105, 2007.
8 A. Caris, G.K. Janssens. *A Local search heuristic for the pre- and end-haulage of intermodal container terminals*. Computers and Operations Research, 36, 2763-2772, 2009.
9 R. Namboothiri and A.L. Erera, *Planning local container drayage operations given a port access appointment system*. Transportation Research Part E: Logistics and Transportation Review, 44, 185-202, 2008.
10 C. Archetti, M. G. Speranza, A. Hertz. *A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem*. Transportation science, 40, 1, 64-73, 2006.
11 C. H. Papadimitriou. *Computational Complexity*. Addison Wesley Longman, 1993.

Table 2 The solutions of artificial instances.

$\|I\|$	$\|E\|$	HEURISTICS $H0$	$H1$	$H2$	$H3$	$H4$	$H5$	$H6$	$H7$	t(s)	% Gap from CPLEX	CPLEX % Optimality Gap
						10 CLIENTS						
2	8	24170.26	20872.07	21392.98	**20672.07**	21392.98	22985.55	22985.55	**20672.07**	6	0.00	4.06
5	5	23674.27	21544.40	20916.21	22411.33	20066.19	20729.28	21131.93	**19960.83**	0	0.00	0.00
8	2	24067.78	23528.75	**21109.77**	23175.59	**21109.77**	21289.41	22863.04	**21109.77**	0	1.95	4.04
						20 CLIENTS						
2	18	46436.64	41566.04	43639.35	**40766.04**	43639.35	43319.77	43315.09	**40766.04**	0	-2.93	7.51
5	15	46204.46	36873.01	42743.03	**36233.01**	41845.95	40636.38	40378.63	**36233.01**	0	-4.48	7.91
10	10	46326.75	34751.04	36256.89	35277.56	36608.49	35976.40	38127.17	**33092.29**	0	4.20	4.40
15	5	46204.46	40021.52	37336.19	40222.93	38196.39	40480.64	42463.88	**37261.67**	0	-1.08	6.29
18	2	46204.46	44019.68	43050.79	44877.00	43050.79	44459.09	44753.25	**42975.11**	42	-2.41	7.02
						30 CLIENTS						
2	28	69121.47	63708.18	66750.74	**62728.18**	66730.74	66871.79	66682.33	**62728.18**	3	-6.38	9.52
5	25	69906.97	60019.63	66365.54	59423.33	65468.45	64809.86	66863.83	**59359.63**	2	-1.76	6.66
10	20	70704.03	54774.37	60574.18	54740.43	59587.76	58581.67	64047.55	**54129.24**	1	-5.63	14.27
15	15	70905.38	52092.61	50934.97	52687.38	53340.51	51172.63	60816.68	**49218.46**	1	-4.44	10.37
20	10	70905.38	59170.97	55004.74	60576.35	55574.39	58998.61	63808.75	**54757.12**	1	n.s	n.s.
25	5	69481.27	66695.13	61075.33	66554.36	61075.33	64538.63	65977.58	**61074.67**	2	-13.11	15.51
28	2	68982.30	69906.35	**67217.27**	68563.65	**67217.27**	68361.82	68361.82	**67217.27**	3	-5.17	9.93
						40 CLIENTS						
2	38	101302.11	95645.86	99695.34	**93985.86**	100062.51	96291.21	96952.05	**93985.86**	17	n.s	n.s.
5	35	101852.55	88764.09	91654.05	**88244.09**	92053.54	95498.95	99401.32	**88224.09**	11	n.s	n.s.
10	30	101582.68	76194.52	82999.48	75968.82	85442.09	86621.65	90626.66	**75117.98**	49	n.s	n.s.
15	25	101776.10	70920.95	76016.82	70347.44	75518.30	74980.22	89368.12	**68484.46**	3	n.s	n.s.
20	20	101895.27	76525.84	71514.12	76941.13	71709.14	75641.02	89044.40	**70795.53**	2	n.s	n.s.
25	15	102349.92	83733.24	76749.41	83267.22	76805.40	87505.43	95387.34	**76261.72**	5	n.s	n.s.
30	10	101286.43	91838.75	85772.27	92560.11	85888.37	91851.87	95243.18	**85506.14**	6	n.s	n.s.
35	5	101735.75	99057.56	**95072.84**	97828.28	**95072.84**	98095.70	100364.94	**95072.84**	11	n.s	n.s.
38	2	100722.47	100265.05	**97736.69**	99516.60	**97736.69**	100109.32	100109.32	**97736.69**	11	n.s	n.s.
						50 CLIENTS						
2	48	132095.20	125233.97	130617.35	**124513.97**	130761.45	130857.73	130959.03	**124513.97**	20	n.s	n.s.
5	45	131432.46	120617.96	126258.08	**120197.96**	125786.41	125447.20	128393.50	**120197.96**	23	n.s	n.s.
10	40	132098.19	113127.96	116088.33	**112687.96**	116348.43	120467.73	124113.76	**112687.96**	15	n.s	n.s.
15	35	133459.72	102304.67	113048.46	**100770.48**	113085.01	111324.61	121595.94	100837.89	8	n.s	n.s.
20	30	132673.52	94279.48	98966.22	92089.72	103620.29	99575.14	113407.89	**91542.11**	5	n.s	n.s.
25	25	131859.53	84807.36	91318.81	83254.21	92807.30	86028.26	112155.51	**82373.34**	5	n.s	n.s.
30	20	131907.73	89854.06	86239.26	88105.14	86865.49	90237.42	111915.47	**83923.10**	6	n.s	n.s.
35	15	132079.70	103480.43	96405.91	104061.17	96672.63	106294.63	117055.94	**95771.71**	8	n.s	n.s.
40	10	131450.60	114221.90	106093.40	116702.34	106920.76	115479.38	124521.98	**105976.92**	10	n.s	n.s.
45	5	132793.70	123151.73	113013.98	122472.64	114208.57	123625.22	124787.33	**112704.72**	19	n.s	n.s.
48	2	132892.88	127984.28	**122462.37**	126229.39	**122462.37**	128170.93	130264.79	**122462.37**	35	n.s	n.s.

Optimization of electricity trading using linear programming*

Minja R. Marinović[1], Dragana D. Makajić-Nikolić[1], Milan J. Stanojević[1], and Lena S. Đorđević[1]

1 **Faculty of organizational sciences, University of Belgrade**
Jove Ilića 154, Belgrade, Serbia
marinovic75709@fon.bg.ac.rs, gis@fon.bg.ac.rs, milans@fon.bg.ac.rs, lena.djordjevic@fon.bg.ac.rs

Abstract

In the last two decades, the liberalization of the electricity markets have been established in order to increase efficiency, harmonize and reduce electricity prices, make a better use of resources, give customers the right to choose their supplier and provide customers with a better service. This change made the electricity market competitive and introduced several new subjects. In this paper, we study one of these subjects: Electricity Trading Company (ETC) and its daily trading process. We present a linear mathematical model of total daily profit maximization subject to flow constraints. It is assumed that the demand and supply are known and some of them are arranged. Possible transmission capacities are known but also additional capacities can be purchased. All trading, transmission prices and amounts are subject of auctions. First, we present energy trading problem as directed multiple-source and multiple-sink network and then model it using linear programming. Also, we provide one realistic example which is slightly changed in order to save confidentiality of the given data.

1998 ACM Subject Classification G 1.6 Optimization, G 2.2 Network Problems

Keywords and phrases energy daily trading, day-ahead planning, LP model, optimization

Digital Object Identifier 10.4230/OASIcs.SCOR.2012.94

1 Introduction

Until the nineties, power plants power production and transmission were carried out between monopolistic public power companies. This situation changed since electricity markets have been deregulated allowing customers to choose their provider and producers. Today, we have a Single European electricity market which provides seamless competition within the electricity supply chain. All participants enjoy a wide choice between competing electricity retailers, who source their requirements in competitive wholesale markets. Market participants are trying to satisfy demand in their own countries and supply electricity across borders into neighborhood markets. Cross-border trading and supply is a part of this market. Energy Trading Companies (ETCs) are buying transmission capacity from Transmission System Operators (TSOs) [1]. All time, TSOs consistently release to the market a truly maximum amount of cross-border transmission capacity. We should mention that ETCs are operating in the middle stage of an energy supply chain [11] and they are trying to manage the risks associated with fluctuating prices through buying and selling electricity contracts. Both

* This research was partially supported by the Ministry of Education and Science, Republic of Serbia, Projects numbers TR32013 and TR35045.

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn; pp. 94–102
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

traders and end-users apply financial instruments such as futures, options and derivatives to protect their exposures to prices and to speculate on price fluctuations.

There are few ways of trading electricity but two main ways are via the telephone in bilateral transactions (so called "Over The Counter" or OTC, usually through the intermediation of a broker), or it is traded through futures markets such as Nordpool or EEX. Some key factors influencing energy prices include geopolitical factors, global economic growth, short term weather impacting demand, supply disruptions from maintenance or unexpected outages, fuel price movements and product swapping in response to relative prices [3].

The literature concerning different issues in energy trading and transmission is extensive. In [8] the authors analyze the impact of CO2 cost on power prices and test two different average cost pricing policies, regional and zonal, that have different effects on electricity market of Central and West Europe. In [12] multiple interrelated markets is considered for electricity and propose a multi-stage mixed-integer stochastic model for capacity allocation strategy in a multi-auction competitive market. A generalized network flow model of the national integrated energy system that incorporates production, transportation of coal, natural gas, and electricity storage with respect to the entire electric energy sector of the U.S. economy is proposed in [10]. The authors have formulated a multi period generalized flow problem in which the total cost is minimized subject to energy balance constraints. The problem of energy allocation between spot markets and bilateral contracts is formulated as a general portfolio optimization quadratic programming problem in [6]. The proposed methodology with risky assets can be applied to a market where pricing, either zonal or nodal, is adopted to mitigate transmission congestion. In [9] the authors propose a zonal network model, aggregating individual nodes within each zone into virtual equivalent nodes, and all cross-border lines into equivalent border links. Using flow-based modeling, the feasibility of the least granularity zonal model where the price zones are defined by the political borders, is analyzed. The authors in [4] consider network systems with multiple-source multiple-sink flow such as electric and power systems. They observe the problem in which resources are transmitted from resource-supplying nodes to resource-demanding nodes through unreliable flow networks. In [7] they analyze the simultaneous optimization of power production and day-ahead power trading and formulated it as a stochastic integer programming model.

In this paper, we consider day-ahead planning from the perspective of a decision maker in ETC. Decisions that should be made are: where and how much electricity ETC should buy and sell and on which ways that energy should be transferred in order to maximize total daily profit. We represent this problem by a directed multiple-source and multiple-sink network and model using linear programming. We illustrate this approach on an example of one ETC trading on Central and South-East Europe (CSEE).

The paper is organized as follows. Section 2 is devoted to a description of the main assumptions of the observed problem. In Section 3, we present a LP model for day-ahead planning. Then, in Section 4 we report and discuss numerical results to illustrate the model application. Also, we present the solution of the LP model with fixed parameters and then we investigate the impact of prices on trading capacities and amounts. Conclusions along with perspectives regarding further work are given in Section 5.

2 Problem description

In this paper we are focusing on electricity trading from the perspective of ETC. The main task of trading section of ETC is to ensure each client's demands are met whatever the circumstances. Trading section also enables ETC to respond to the ever-changing state of the

Figure 1 Simplified CSEE electricity market.

region's transmission grid and production capacities. Besides that, this section takes care of different problems such as: spot and long term arrangements, making schedules, cross border capacities allocations, optimization of whole portfolio, managing different energy sources, customers in different countries and cross border energy flows and costs.

Efficiency of trading section of ETC can be improved by dealing with at least two optimization problems: long term and short term (day-ahead) planning. Long term planning include determination of electricity market e.g. buyers and suppliers interested for cooperation and transmission capacity that will be purchased for the next period. Day-ahead planning represents finding the optimal plan of selling and buying electricity that will maximize daily profit considering the available transmission capacities.

Day-ahead planning starts from an established network of potential buyers and suppliers and purchased transmission capacities based on long term decisions. One example of simplified electricity network where all buyers and suppliers from one country are represented by one node is shown in Figure 1. Similar network will be used later in the numerical example.

Daily demand and daily supply of each country (node) are known. Some electricity trades are already arranged and they must be fulfilled. All the electricity that has been bought during one day has to be sold the same day. Therefore, if there is a surplus or shortage of arranged supply, it will be traded through future markets.

In order that transmission capacities purchased for a long period to be disposed by ETC, it is necessary to announce the amount that will be used. Only the amounts that are announced one day before day-ahead planning are considered available. The transmission capacity that is not announced is subject to the "use it or lose it" principle and will be reoffered in the daily auction [5]. If daily trading exceeds the amount of announced capacity, it is possible to buy additional daily transmission capacities at the auction price.

The goal is to create a useful tool which will help decision makers in trading section of ETC to simulate market and network situations for different amounts and prices of electricity and transmission capacities in order to help them to maximize ETC's total profit.

3 Model formulation

The described problem can be modeled as a directed multiple-source and multiple-sink network. In this section, we will present notation for this problem which will be used for the LP mathematical model for day-ahead planning. Sets N, S, B and A represent: set of all nodes, set of all nodes representing sellers ($S \subseteq N$), set of all nodes representing buyers ($B \subseteq N$) and set of arcs representing transmission capacity between nodes ($A \subset N \times N$), respectively. Parameter su_j is the upper bound of electricity that can be bought from supplier j, for every $j \in S$ while sl_j represents the lower bound (arranged buying) of electricity that must be bought from supplier $j \in s$. Also, the upper bound bu_i and the lower bound bl_i (arranged sale) of electricity that can be sold, are given for each buyer i, where $i \in B$. Parameter f_{ij} represents announced daily transmission capacity of every arc $(i,j) \in A$. The maximal additional daily transmission capacity which is possible to buy on an arc $(i,j) \in A$ is denoted by hu_{ij}.

Parameters c_j, d_i, a_{ij} and t_{ij} represent: purchasing prices for supplier $j \in S$, selling price for buyer $i \in B$, price for additional transmission capacities for every arc $(i,j) \in A$ and taxes for additional transmission capacities on arc $(i,j) \in A$, respectively. In this model we have three variables:

- x_j – amount of electricity that should be bought from supplier for every $j \in S$;
- y_i – amount of electricity that should be sold to buyer for every $i \in B$;
- h_{ij} – amount of additional transmission capacities on arc for every $(i,j) \in A$.

We should mention that all electricity amounts and transmission capacities are expressed in MWh. The unit for all prices and taxes is Euro per MWh.

Using given notation, the LP mathematical model for day-ahead planning can be stated as:

(MMDAP)

$$\max Z(x,y,h) = \sum_{i \in B} d_i y_i - \sum_{j \in S} c_j x_j - \sum_{(i,j) \in A} (t_{ij} + a_{ij}) h_{ij} \tag{1}$$

s.t.

$$\sum_{j \in S} x_j - \sum_{i \in B} y_i = 0 \tag{2}$$

$$\sum_{(i,j) \in A} (f_{ij} + h_{ij}) - \sum_{(j,i) \in A} (f_{ji} + h_{ji}) = \begin{cases} 0 & j \in N \backslash (S \cup B) \\ -x_j & j \in S \backslash B \\ y_j & j \in B \backslash S \\ y_j - x_j & j \in B \cap S \end{cases} \tag{3}$$

$$sl_j \leq x_j \leq su_j, \; j \in S \tag{4}$$

$$bl_i \leq y_i \leq bu_i, \; i \in B \tag{5}$$

$$0 \leq h_{ij} \leq hu_{ij}, \; (i,j) \in A \tag{6}$$

The objective function (1) represents the difference between total income and expenses of buying electricity and additional transmission capacities. The constraint (2) provides that

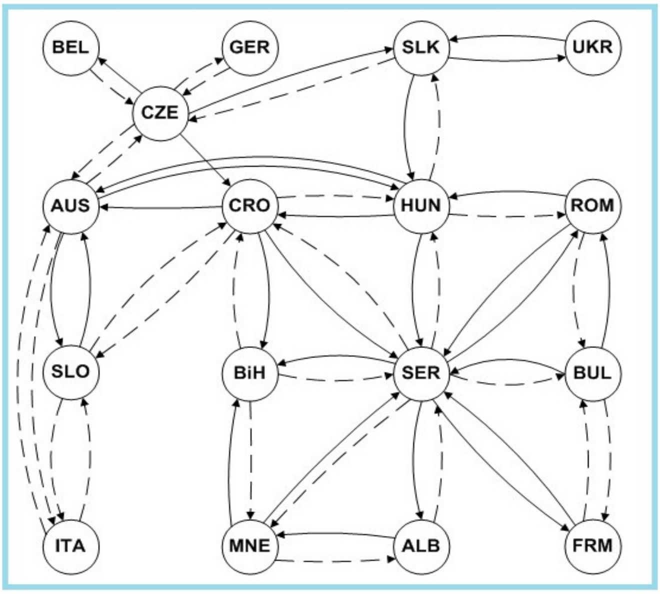


Figure 2 Graph representation of CSEE electricity market.

the sums of electricity that has been bought and sold during one day are equal. Since all nodes are transition nodes, the flow conservation constraint (3) must hold. This constraint has four different interpretations depending on the node type. $N \backslash (S \cup B)$ is the set of nodes without demand and supply. For each node from $N \backslash (S \cup B)$ constraints (3) ensure that the amount of electricity entering node must be equal to the amount of electricity leaving node. $S \backslash B$ is the set of source nodes in which the amount of electricity entering the node and electricity bought in this node must be equal to the amount of electricity leaving this node while $B \backslash S$ represents the set of sink nodes in which the amount of electricity entering a node must equal the amount of electricity leaving a node and electricity sold in this node. $B \cap S$ is the set of source-sink nodes. For each node from $B \cap S$ constraints (3) ensure that the amount of electricity entering the node and electricity bought in this node must equal the amount of electricity leaving this node and electricity sold in that node. Optimal amounts of electricity that should be bought and sold lies between their upper and lower boundaries given by constraints (4) and (5). Constraint (6) refers to maximal additional daily transmission capacity which is possible to buy on an arc.

4 Numerical examples

In order to evaluate the proposed model we consider two different ways of its application. First one is related to obtaining the optimal day-ahead plan when all parameters have fixed values. The second application is based on a more realistic situation when some parameters may vary within certain boundaries. Both applications are demonstrated on the CSEE network consisting of 17 nodes and 54 arcs (Figure 2).

Each country is presented by one node, which is characterized by its lower and upper bounds of electricity that can be bought and/or sold in/from that country as well as purchasing and selling prices. All arcs in Figure 2 (solid and dashed lines) represent cross-border connections on which it is possible to buy additional transmission capacity.

Table 1 Optimal solution.

	Buying in MWh			Selling in MWh		
node	min	Optimal	max	min	Optimal	max
BEL	18	40	40	25	31	67
GER	2	70	70	0	70	80
AUS	1	31	31	0	0	60
ITA	0	36	50	31	41	41
SLO	–	–	–	5	5	35
CZE	25	35	35	0	0	38
HUN	0	0	10	–	–	–
CRO	–	–	–	22	22	44
BiH	36	52	52	42	49	49
MNE	–	–	–	5	5	60
SLK	–	–	–	12	12	33
SER	0	46	46	0	80	80
ALB	5	10	35	–	–	–
FRM	0	40	40	3	3	57
ROM	–	–	–	4	38	70
BUL	–	–	–	14	14	25
UKR	10	10	20	–	–	–

Arcs represented by solid lines indicate the existence of announced transmission capacities purchased earlier. Maximal additional daily transmission capacities, amounts of announced transmission capacities, prices and taxes of each arc are also given. Due to confidentiality issues, we present in this paper slightly modified data which are within boundaries of common real-life situations.

The model has been implemented and solved using the GNU Linear Programming Kit [2]. The optimal solution for one scenario is given in Table 1. Marks "–" mean that there were no suppliers or buyers in the corresponding country.

In addition to the optimal trade amounts, the solution determines additional transmission capacities that should be purchased. As well, the solution provides information about total trade amounts in MWh and profit which will be made with this scenario.

Beside information about total daily trade and total daily income, the application has the possibility to show expenses of buying electricity and expenses of additional transmission capacities in Euros. In order to obtain a total profit certain corrections in the objective have to be made:

- Taxes for announced transmission capacities as well as costs of long term purchased transmission capacities evaluated on daily basis should be subtracted.
- Unit prices for previously arranged buying and selling (bl_j and sl_j) can differ from actual prices. These differences should be taken into account.

However, any of these corrections will not affect the optimal solution obtained by the model. In the further analysis we will use the term "profit" to refer to the value in (1), although it is just an approximation.

In order to demonstrate the second way of using MMDAP we perform a sensitivity analysis for some parameters. In this case we present a scenario where the selling price for node SER varies from 80 to 88 Euro/MWh. We analyze the influence of the selling price on the optimal

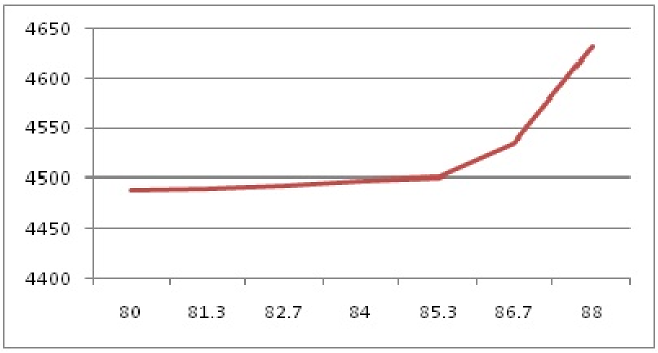


Figure 3 The impact of prices on the optimal profit.

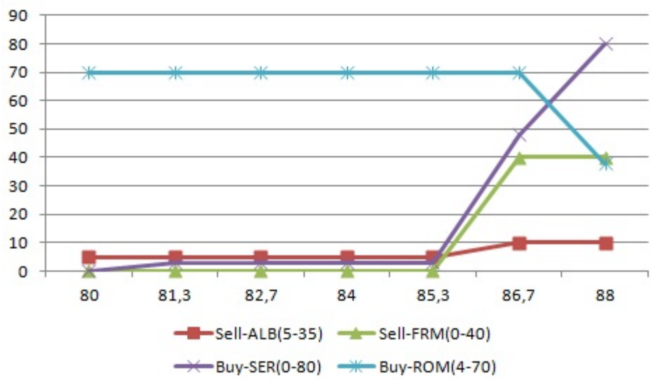


Figure 4 The impact of prices on the optimal amounts.

amount of electricity that should be bought and the optimal solution as a whole. A series of optimizations were made for all integer values of the price in the given interval. Figure 3 and 4 show the optimal amounts of electricity that should be bought and sold for some nodes and the corresponding profits for some characteristic prices of electricity, assuming that all other parameters remain the same.

The increases of the optimal amount of electricity as well as the profit were expected because the selling price increased, but all the solutions are optimal for those values. In other words, for different prices different amounts of electricity that should be bought (and corresponding profits) will be optimal.

Optimization results should give the decision maker information how to make bids on auctions in order to maximize ETC's total profit. On CSEE electricity market uniform price auction type is used. That assumes that each bidder in the auction bids a price and an amount. The price bid is considered the maximum price they are willing to pay per item, and the amount is the number of units they wish to purchase at that price. Typically these bids are sealed – not revealed to the other buyers until the auction closes. The auctioneer then serves the highest bidder first, giving them the number of units requested, then the second highest bidder and so forth until the supply of the commodity is exhausted. All bidders then pay a per unit price equal to the lowest winning bid (the lowest bid out of the buyers who actually received one or more units of the commodity) – regardless of their actual bid [13].

In practice, for each offer bidders make several bids with different amounts and prices in order to provide both: needed amount to be bought and the price to be as low as possible.

Table 2 List of bids derived from optimal solutions.

Bids	Amounts of el.	Price	Accumulated amount	Guaranteed profit
B1	3	481.3-85.3	3	4489-4501
B2	45	85.4-86.7	48	4535
B3	35	86.8-88	80	4631

On the basis of the presented optimal solutions we can suggest the ETC decision maker to make bids as presented in Table 2.
Depending on the price equal to the lowest winning bid (which will not be known till the end of the auction), ETC will buy 0, 3, 48 or 80 MWh from node SER. For prices that are presented as intervals in Table 2, ETC decision maker can give any price from that interval for which it estimates that can win. Anyhow, the corresponding amount will be optimal. The profits given in the rightmost column represent the optimal profit in case that corresponding bid becomes the lowest winning bid. If any lower bid wins, the profit will be bigger.

5 Conclusions

In this paper, problem of day-ahead energy planning in the Energy Trading Companies have been considered. It has been formulated as a linear programming problem. Day-ahead energy planning implies finding the optimal amounts of electricity that should be bought from each supplier and sold to each buyer and the optimal routes which can satisfy the daily demands using the purchased and additional energy transmission capacities.

A real numerical example was considered and demonstrate the usage of the model which has been presented in this paper. Since the developed model is linear, it can be used to solve real life problems of large dimensions.

As a topic of further research, the trading through futures markets can be taken into consideration. Another topic of future research is modeling long term planning strategy of Energy Trading Companies.

References

1 EFET, Electricity markets harmonisation – New edition. http://www.efet.org
2 GLPK, GNU Linear Programming Kit. `http://www.gnu.org/s/glpk/`
3 HARCOURT, Harcourt strategy sheet – Energy and Power Trading. `http://www.harcourt.ch`
4 Hsieh C. C., Chen Y. T. *Resource allocation decisions under various demands and cost requirements in an unreliable flow network*. Comput Oper Res 32:2771–2784, 2005.
5 Kristiansen T. *An assessment of the Danish–German cross-border auctions*. Energy Policy 35:3369–3382, 2007.
6 Liu M., Wu F. F. *Portfolio optimization in electricity markets*. Electr Pow Syst Res 77:1000–1009, 2007.
7 Nowak M. P., Schultz R. D., Westphalenm M. A. *Stochastic Integer Programming Model for Incorporating Day-Ahead Trading of Electricity into Hydro-Thermal Unit Commitment*. Optim Eng 6:163–176, 2005.
8 Oggioni G., Smeers Y. *Evaluating the impact of average cost based contracts on the industrial sector in the European emission trading scheme*. Cent Europ J Oper Re 17, 2:181–217, 2009.

9 Purchala K., Haesen E., Meeus L., Belmans R. *Zonal network model of European interconnected electricity network*, CIGRE/IEEE PES, International Symposium New Orleans, pp. 362–369, 2005.

10 Quelhas A., Gil E., McCalley J. D., Ryan S. M. *A Multiperiod Generalized Network Flow Model of the U.S. Integrated Energy System: Part I–Model Description*. IEEE T Power Syst 22, 2:829–836, 2007.

11 Thery R., Zarate P. *Energy planning: a multi-level and multicriteria decision making structure proposal*. Cent Europ J Oper Re 17, 3:265–274, 2009.

12 Triki C., Beraldi P., Gross G. *Optimal capacity allocation in multi-auction electricity markets under uncertainty*. Comput Oper Res 32:201–217, 2005.

13 Vijay K., *Auction theory*. Academic Press, San Diego, 2002.

A Bilevel Mixed Integer Linear Programming Model for Valves Location in Water Distribution Systems

Andrea Peano, Maddalena Nonato, Marco Gavanelli, Stefano Alvisi, and Marco Franchini

Department of Engineering,
University of Ferrara
Via Saragat, 1
44122, Ferrara, Italy
{andrea.peano, maddalena.nonato, marco.gavanelli, stefano.alvisi, marco.franchini}@unife.it

Abstract

The positioning of valves on the pipes of a Water Distribution System (WDS) is a core decision in the design of the isolation system of a WDS. When closed, valves permit to isolate a small portion of the network, so called a sector, which can be de-watered for maintenance purposes at the cost of a supply disruption. However, valves have a cost so their number is limited, and their position must be chosen carefully in order to minimize the worst-case supply disruption which may occur during pipe maintenance. Supply disruption is usually measured as the undelivered user demand. When a sector is isolated by closing its boundary valves, other portions of the network may become disconnected from the reservoirs as a secondary effect, and experience supply disruption as well. This induced isolation must be taken into account when computing the undelivered demand induced by a sector isolation. While sector topology can be described in terms of graph partitioning, accounting for induced undelivered demand requires network flow modeling. The aim of the problem is to locate a given number of valves at the extremes of the network pipes so that the maximum supply disruption is minimized. We present a Bilevel Mixed Integer Linear Programming (MILP) model for this problem and show how to reduce it to a single level MILP by exploiting duality. Computational results on a real case study are presented, showing the effectiveness of the approach.

1998 ACM Subject Classification G.1.6 Optimization

Keywords and phrases Isolation Valves Positioning, Bilevel Programming, Hydroinformatics

Digital Object Identifier 10.4230/OASIcs.SCOR.2012.103

1 Introduction

In this Section we introduce a real problem in hydraulic engineering concerning the location of the isolation valves of a Water Distribution System, and reformulate it as a graph based optimization problem. A mathematical model is presented in Section 2, computational results are presented in Section 3 where conclusions are drawn.

1.1 Valves closure and sector isolation

Water Distribution Systems (WDSs) are complex systems whose mission is to supply water to the communities living in their service area. A WDS is made of several components, the main ones being: a set of reservoirs feeding the WDS, a set of pipes delivering water to the

3rd Student Conference on Operational Research (SCOR 2012).
Editors: Stefan Ravizza and Penny Holborn; pp. 103–112
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

system users, a user demand for each pipe, describing the average water consumption by the users served by that pipe (liters per second), a set of junctions each describing the connection of two or more pipes to each other.

Users are connected to their closest pipe by way of smaller pipes through which water is supplied. To this purpose, we can imagine users as being evenly distributed along the pipe. In turn, pipes receive water at an adequate pressure from the reservoirs to which they are connected in the hydraulic network. Usually, the topological layout of hydraulic networks contains a few loops that increase network reliability. Thus, a pipe can be connected to a reservoir by several different paths. Given a pipe, if each connection from the pipe to each reservoir is interrupted, water pressure falls, the pipe no longer supplies its users and it is said to be *isolated*. Failure of ageing pipes frequently occurs. In such a case, the leaking pipe is isolated on purpose, to be dewatered and fixed. Isolation is achieved by closing some of the isolation valves purposely located on the network, in such a way that the failed pipe gets disconnected from the reservoirs. In an ideal situation, each pipe would have one such valve positioned at each of its two extremes, so that only that pipe could be disconnected in case of maintenance by closing just its two valves, but it would require twice as many valves as the network pipes. However, the number of valves is limited due to cost, and their location poses a challenge, as described hereafter.
First, valves must be properly located at pipe extremes in such a way that any pipe can be isolated by closing a proper set of valves. When all valves are closed, the network is subdivided into a set of subnet, called *sectors*; also, the valves that delimit a sector are called the *boundary valves* of that sector. Said in another way, a sector is a set of pipes which stay connected themselves once all valves are closed. It follows that pipes within the same sector share the same status, either isolated or connected to a reservoir, depending on which valves are closed. When a sector is isolated, all its users experience supply disruption.
Second, the WDS engineers who design the network aim to reduce and equally distribute the service disruption among users in case of maintenance operations.

So far, if we suppose that each pipe is equally likely to fail and require maintenance, this target would be achieved by any valves location yielding sectors whose user demand is approximately the same. However, a secondary effect of sector isolation must be accounted for, i.e., *unintended isolation*. A pipe for which all connections to the reservoirs go through the isolated sector will be isolated as well when that sector is closed. Therefore, the supply disruption associated to a sector cannot take into account only the user demand of the sector itself, but must consider the demand of unintentionally isolated pipes as well. We illustrate these concepts on the toy network depicted in Figure 1. The hydraulic network has a single reservoir, 6 junctions and 7 pipes with positive demand, plus a 0-demand pipe which connects the reservoir to the rest of the network. 5 valves are positioned at pipe extremes as follows: near junction 1 on pipe $(1, 2)$, near junction 1 on pipe $(1, 4)$, near junction 2 on pipe $(2, 3)$,

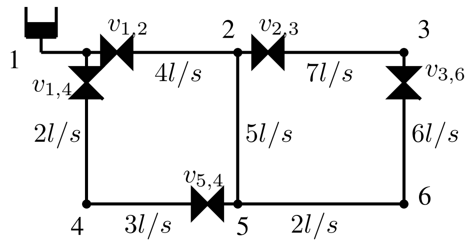


Figure 1 A simple hydraulic network and its isolation system.

near junction 3 on pipe $(3, 6)$, and finally near junction 5 on pipe $(4, 5)$. Three sectors are induced by those valves, namely S_1 made of pipes $(1, 4)$, $(4, 5)$ with demand $5l/s$; S_2 made of pipes $(1, 2)$, $(2, 5)$, $(5, 6)$, $(3, 6)$ with demand $17l/s$; S_3 made of pipe $(2, 3)$ with demand $7l/s$. Let $v_{i,j}$ denote a valve located on pipe (i, j) near i, and let $v_{j,i}$ be the one close to j. If pipe $(2, 3)$ needs repairing, valves $v_{1,2}$ and $v_{3,6}$ can be closed, with a supply disruption of $7l/s$. If pipe $(5, 6)$ leaks, the boundary valves of S_2 will be closed, namely $v_{1,2}$, $v_{5,4}$, $v_{3,6}$, and $v_{2,3}$, but the supply disruption will be $24 = 17 + 7$, greater than its sector demand, accounting for the unintended isolation of S_3.

The Isolation System Design (ISD) of WDSs consists of locating a limited number of valves at pipe extremes so that any pipe can be isolated. What an optimal placement is may depend on several criteria that give rise to different objective functions; in particular, the *Bottleneck Isolation Valves Location Problem* (BIVLP) minimizes the maximum undelivered demand. The two main issues related to the ISD problem addressed in the hydraulic engineering literature, are: the identification of the segments and unintended isolation due to the closure of some isolation valves, and the optimal location of isolation valves. Regarding the first topic, among others, [12, 13] exploit a dual representation of the network, with segments treated as nodes and valves as links; [5, 9] exploit the topological incidence matrices to identify the segments. Regarding the second topic, both [9] and [5] tackle the problem by bi-objective genetic algorithms, seeking a compromise between cost and solution quality. In particular, the former minimizes the number of valves and the maximum supply disruption. The latter minimizes the cost of the installed valves related to pipe diameters and the average supply disruption weighted by the probability of pipe failure. Both provide an approximation of the Pareto frontier and no bounds to evaluate the quality of the heuristic solutions.

The only two exact approaches for the BIVLP we are aware of are provided in [2] and [7], which apply two different tools of the Logic Programming field to minimize the maximum undelivered demand given a fixed number of valves. The former is based on Constraint Logic Programming on Finite Domain [11] (CLP(FD)), and models the problem as a two-players game and three moves: player 1 locates the valves, player 2 chooses a pipe to break, and player 1 closes a set of valves. The latter is based on the Answer Set Programming [8] (ASP) paradigm and uses the concept of "*extended sector*", that is the union of a sector with its (if existing) unintended isolations. Both can be used to compute the Pareto frontier of the problem tackled by [9] by solving a sequence of instances with an increasing number of valves. Different models for the BIVLP have been proposed in each paper, with different pros and cons regarding prototyping and computing times. At present the best computational performance is achieved by the CLP(FD) model [2], which implements a redundant valves elimination technique and bounding procedures, and which is our benchmark.

In the following we reformulate the BIVLP in the framework of graph theory, and set the basis for the mathematical optimization model provided in Section 2.

1.2 Hydraulic sectors and graph partitioning

The Graph Partitioning problem (GPP) is one of the most studied problems in combinatorial optimization, and admits several variants. Recall that a partition of a set is a collection of non-empty disjoint subsets whose union returns the set. Generally speaking, GPP consists of partitioning the vertices of a graph into a set of connected components by removing a minimum (weight) set of edges, according to some criteria. For example, the number of components of the partition is fixed, and the number of nodes in each component is bounded from above. The set of removed edges is usually referred to as a *multicut*. Literature references are many, among which [3] investigates the polytopes associated to the integer programming formulations of the major variants, and [6] studies the convex hull of incidence vectors of feasible multicuts for the capacitated GPP.

The several analogies between the hydraulic network sectorisation and graph partitioning, as recently exploited in the hydraulic engineering literature [14], seem at first to provide an ideal framework for modelling BIVLP. Recall that each pipe must belong to one sector in order to be isolated if required, but only one sector, due to the minimality of sector definition. Therefore, sectors induce a partition on the positive demand pipes, whose associated multicut models the location of the isolation valves at pipe extremes. This simple correspondence, though, does not allow us to address the issue of unintended isolation. To meet this requirement we need to introduce an extended graph that represents the topology of the network, on which flow variables will model water flows, thus capturing unintended supply disruption. We first provide a graph representation of the hydraulic network supporting graph partition, and in the next Section we show how to extend such graph to handle unintended isolation.

First, we suppose to concentrate the user demand of a pipe (i, j) at a single point denoted as ϵ_{ij}, located in the middle of the pipe. Denote by $\delta_{\epsilon_{ij}}$ such demand. Then, we shrink all the reservoirs into a single one, to which we associate node σ. Now we can introduce the undirected graph $G = (V, E)$ defined as follows.
The vertex set V is made of the union of: $\Sigma = \{\sigma\}$, where σ is the super source modelling the set of reservoirs; Ψ, being the union over all pipes of the demand points ϵ_{ij}; Γ, modelling the junctions of the hydraulic network.
The edge set E is made of: T, the set of the few structural edges corresponding to 0-demand pipes which connect σ to a vertex in Γ; we denote as $\Gamma(\Sigma) \subset \Gamma$ the subset of the junction nodes adjacent to a reservoir. F, made of a pair of edges (i, ϵ_{ij}) and (j, ϵ_{ij}) for each pipe (i, j) in the hydraulic network. Figure 2 shows how the two adjacent pipes of the hydraulic network are modelled in graph G.

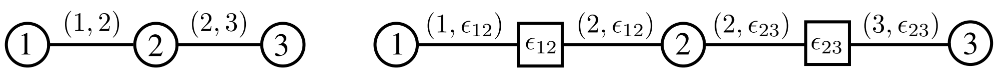


Figure 2 Two adjacent pipes of the hydraulic network and their representation in G.

Each edge in F may host a valve, while edges in T do not. In real life WDSs, actually, a special valve is always present on each pipe in T in order to isolate the reservoir from the network, if the reservoir needs maintenance. However, such a valve would never be closed for pipe maintenance purposes, so we disregard it here. Therefore, we assume that the edges in T bear no valves, and that these edges, the reservoirs and the nodes in $\Gamma(\Sigma)$ do not belong to any sector. Furthermore, to keep notation simple, we suppose that all pipes other than those incident on the reservoirs have positive demand.

Let h_{max} be the number of available valves, and s_{max} the maximum number of sectors admitted. Denote by Δ_s, for $s \in S = \{1, .., s_{max}\}$, the undelivered demand associated to sector s, and it is given by the sum of $\delta_{\epsilon_{ij}}$ for each pipe (i, j) which gets isolated when the boundary valves of sector s are closed. Note that isolating a sector s on the hydraulic network corresponds to the removal of all edges $(i, \epsilon_{ij}) \in F$ on graph G such that a boundary valve of s is positioned on pipe (i, j) near junction i. We search for the location of at most h_{max} valves on as many edges of F yielding at most s_{max} sectors such that $\max_{s \in S}\{\Delta_s\}$ is minimum. In Section 2 we provide a mathematical model for this problem.

2 Mathematical models for the BIVLP

In 2.1 we present a graph partitioning model on graph G, how to extend G to model unintended undelivered demand, and a bilevel MILP model for BIVLP. In 2.2 it is presented how to reduce the bilevel model to a single level MILP.

2.1 A Bilevel model for BIVLP

Let us introduce the following variables.

τ_{ij}^s $\in \{0,1\}$ $\forall (u,v) \in F : u = i, v = \epsilon_{ij}$, $\forall s \in S$, is a binary variable equal to 1 if a boundary valve for s is located on pipe (i,j) near i in the WDN (edge (i,ϵ_{ij}) in G), and 0 otherwise. Likewise, $\tau_{ji}^s = 1$ if the valve is near j, that is, on edge (j,ϵ_{ij}) in G.

z_i^s $\in \{0,1\}$ $\forall i \in \Psi \cup \Gamma \setminus \Gamma(\Sigma)$, $\forall s \in S$, is a binary variable equal to 1 if node i belongs to sector s and 0 otherwise.

Recall that variable z_i^s is not defined for $i \in \Gamma(\Sigma)$ or $i = \sigma$. If $\tau_{ij}^s = 1$ then there must be another sector s' such that $\tau_{ij}^{s'} = 1$, unless i is adjacent to a reservoir. Indeed, one valve separates (at most) two sectors from each other.

The following constraints partition the vertices of G into at most s_{max} sectors with limited user demand, by cutting at most h_{max} edges in F.

$$\sum_{s\in S} z_i^s = 1, \qquad \forall i \in \Psi \cup \Gamma \setminus \Gamma(\Sigma). \tag{1a}$$

$$\sum_{s\in S} \sum_{\epsilon_{ij}\in\Psi} (\tau_{ij}^s + \tau_{ji}^s) \le h_{max} \tag{1b}$$

$$\sum_{\epsilon_{ij}\in\Psi} z_{\epsilon_{ij}}^s \delta_{\epsilon_{ij}} \le \delta_{max}, \qquad \forall s \in S, \tag{1c}$$

These constraints impose the following. Each node in $\Psi \cup \Gamma \setminus \Gamma(\Sigma)$ belongs to one and only one sector (1a). At most h_{max} valves are available (1b). The sum of user demands of the edges within the same sector is bounded from above; constraint (1c) excludes very unbalanced partitions with sectors with large demand. However, the threshold δ_{max} as well as the parameters h_{max} and s_{max} must be carefully set so that a feasible solution exists and no optimal solution is cut off. Furthermore, note that it is not required to use all the available valves since the optimal solution value does not necessarily improve as the number of boundary valves increases, and we want to avoid solutions with useless valves, i.e., valves positioned on an edge whose vertices belong to the same sector. Actually, in (2a–2b) we exploit this fact and take for granted that each valve is a boundary valve, that is, if a valve is positioned on an edge then the vertices of that edge belong to different sectors.

The following constraint states the relation between a boundary valve and the sector of the vertices of the edge where the valve is located. Recall that τ_{ij}^s refers to a valve positioned in between vertex i and vertex ϵ_{ij}. We use the symbol $\oplus$ to denote the XOR logical operator and then provide the system of integer linear inequalities that define the operator.

$$\tau_{ij}^s = z_i^s \oplus z_{\epsilon_{ij}}^s, \qquad \forall (i,\epsilon_{ij}) \in F,\ i,j \notin \Gamma(\Sigma)\ \forall s \in S \tag{2a}$$

$$\tau_{ij}^s = z_j^s, \qquad \forall (i,\epsilon_{ij}) \in F : i \in \Gamma(\Sigma) \tag{2b}$$

Constraints (2a – 2b) state that a boundary valve of sector s is positioned on an edge if and only if exactly one of the two vertices belongs to s, unless the vertex to which the valve is close is adjacent to a reservoir. (2a) can be formulated as the following set of linear inequalities, for each edge $(i, \epsilon_{ij}) \in F$ and for each $s \in S$.

$$\tau_{ij}^s \leq z_i^s + z_{\epsilon_{ij}}^s \tag{3a}$$

$$\tau_{ij}^s \leq 2 - (z_i^s + z_{\epsilon_{ij}}^s) \tag{3b}$$

$$\tau_{ij}^s \geq z_i^s - z_{\epsilon_{ij}}^s \tag{3c}$$

$$\tau_{ij}^s \geq z_{\epsilon_{ij}}^s - z_i^s \tag{3d}$$

The user demand that is satisfied when sector s is isolated clearly depends on where the sector's boundary valves have been located, according to the current configuration of the isolation system which is expressed in terms of the τ^s variables. Let us denote the satisfied user demand as $DD^s(\tau^s)$ and let $\Upsilon = \sum_{(i,j)} \epsilon_{ij}$ be the total user demand. Therefore, BIVLP can be stated as $min_\tau \Delta$ $s.t.$ $\Delta \geq \Upsilon - DD^s(\tau^s)$, $\forall s \in S$ and τ satisfies (1a–1c), (2b), (3a–3d). In the following, we provide a mathematical description of $DD^s(\tau^s)$ as the solution of an optimization problem.

Now we are ready to extend graph G in order to support the introduction of the flow variables required to model water flow on the network corresponding to the satisfied user demand in case a given sector is isolated. Since sectors are isolated one at a time, we must consider separately each such scenario. In order to compute the undelivered demand for a sector, an associated Maximum Flow Problem (MFP) on a graph whose topology depends on τ must be solved.

We start by adding to V a special node P representing a sink collecting all user demand that is satisfied; accordingly, E is extended to include a new set of edges, say D, that carry user demand from each demand-vertex ϵ_{ij} to the sink P.
Then, let us introduce s_{max} families of multicommodity flow variables, one for each sector. They are used to represent water flows in the hydraulic network when that given sector is isolated.

First, we define the variables modelling the flow on network pipes. For each $s \in S$ and $(u, v) \in F$, a pair of *non negative* flow variables are introduced, namely, x_{uv}^s and x_{vu}^s. Recall that $v = \epsilon_{ij}$ for some pipe (i, j), and $u \in \{i, j\}$. Therefore, each pipe (i, j) yields four flow variables for each sector, namely, $x_{i,\epsilon_{ij}}^s$, $x_{\epsilon_{ij},i}^s$, $x_{j,\epsilon_{ij}}^s$, and $x_{\epsilon_{ij},j}^s$. All such variables are bounded by the sum of users demand Υ, unless a boundary valve for s is located on the edge, say near i. In such a case, $\tau_{ij}^s = 1$ and $x_{i,\epsilon_{ij}}^s = x_{\epsilon_{ij},i}^s = 0$.
Second, we introduce flow variables on the demand edges in D, connecting each demand vertex ϵ_{ij} to the sink P. For each $s \in S$ and $\epsilon_{ij} \in \Psi$, let $x_{\epsilon_{ij},P}^s$ be such variable. This variable, for any s, is bounded form above by the actual user demand of pipe (i, j), that is $\delta_{\epsilon_{ij}}$ previously introduced.
Finally, for each $s \in S$ and for each edge in T connecting the reservoir $\sigma \in \Sigma$ to a junction $\gamma \in \Gamma$, a non negative flow variable $x_{\sigma,\gamma}^s$ is introduced, with no upper bound. In Figure 3 the extended graph is depicted. In order to represent the water flow when a given sector s is isolated, we must solve a MFP from the reservoir σ to the sink P with respect to the flow

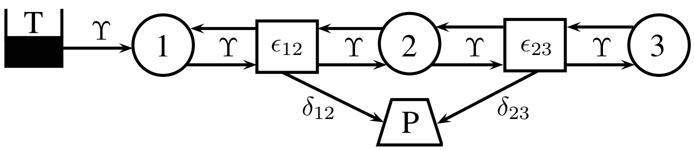


Figure 3 The extended graph. The arrows show in which direction flow can traverse an edge.

variables indexed by s. For each sector s, the mathematical model of the MFP is defined by balance constraints at the nodes, capacity constraints on the flow variables, and the objective function is the maximization of the flow entering P. The optimal value provides the user demand that is satisfied when s is isolated. Capacity goes down to 0 on the arcs where boundary valves of sector s have been located, so the optimal solution value $DD^s(\tau^s)$ is a function of τ^s. Constraints (4–5i) model the MFP for a given sector s. A fake arc going from the sink P to the source σ with non negative flow variable $x^s_{P\sigma}$ is introduced so that the problem can be stated as a circulation problem. Arc (P, σ) is the only arc outgoing from P and entering σ, therefore the objective function can then be stated as maximizing $x^s_{P,\sigma}$. Zero balance constraints at junction nodes are given in (5a), at demand nodes in (5b), at the sink in (5c), and at the reservoir in (5d). Capacity constraints for the flow variables on the edges in F, which depends on valves location, are given in (5e-5f). Similarly, capacity constraints for the flow variables defined on the demand edges in D are given in (5g).

$$DD^s(\tau^s) \quad = max \quad x^s_{P,\sigma} \tag{4}$$

$$s.t. \quad x^s_{\sigma,i} + \sum_{j:\epsilon_{ij}\in\Psi} (x^s_{\epsilon_{ij},i} - x^s_{i,\epsilon_{ij}}) = 0 \qquad \forall i \in \Gamma, \tag{5a}$$

$$x^s_{i,\epsilon_{ij}} + x^s_{j,\epsilon_{ij}} - x^s_{\epsilon_{ij},i} - x^s_{\epsilon_{ij},j} - x^s_{\epsilon_{ij},P} = 0 \qquad \forall \epsilon_{ij} \in \Psi, \tag{5b}$$

$$\sum_{\epsilon_{ij}\in\Psi} x^s_{\epsilon_{ij},P} - x^s_{P,\sigma} = 0, \tag{5c}$$

$$x^s_{P,\sigma} - \sum_{i\in\Gamma(\Sigma)} x^s_{\sigma,i} = 0, \tag{5d}$$

$$x^s_{i,\epsilon_{ij}} \leq \Upsilon(1 - \tau^s_{ij}) \qquad \forall (i, \epsilon_{ij}) \in F, \tag{5e}$$

$$x^s_{\epsilon_{ij},i} \leq \Upsilon(1 - \tau^s_{ij}) \qquad \forall (i, \epsilon_{ij}) \in F, \tag{5f}$$

$$x^s_{\epsilon_{ij},P} \leq \delta_{\epsilon_{ij}} \qquad \forall \epsilon_{ij} \in \Psi, \tag{5g}$$

$$x^s_{\sigma,i} \leq \Upsilon \qquad \forall i \in \Gamma(\Sigma), \tag{5h}$$

$$x^s_{i,j} \geq 0, \qquad \forall (i, j) \in E. \tag{5i}$$

Finally, BIVLP can be stated as a bilevel optimization problem.
$min_\tau \Delta \; s.\; t. \quad$ (1a–1c), (2b), (3a–3d), $\Delta \geq \Upsilon - DD^s(\tau^s) \; \forall s \in S, \quad DD^s(\tau^s) =$ (4-5i).
Regarding the ISD problem, the leader select a feasible configuration of the isolation system by setting τ, while the follower determines the quality of a such configuration by solving a maximum flow problem for each sector.

2.2 A one level MILP for BIVLP

Bilevel optimization [4] provides the framework for modelling optimization problems where two decision makers with conflicting objective functions are involved in a hierarchical relationship. The leader takes its decisions aware of the fact that their value depends on how the follower reacts to such decisions. Here, the leader sets the topology of the isolation system, locating the available valves on the pipes. The follower, sector by sector, maximizes the flow from σ to P on a graph whose topology depends on the boundary valves of the sector. Bilevel optimization problems can be tackled by imposing inner problem optimality by adding its optimality conditions, usually expressed as non linear constraints, to the inner problem feasibility constraints, and reformulating the whole problem as a single level optimization problem. When the inner problem is a linear programming problem, duality can be exploited to state optimality by way of linear constraints (see [1]). In our case, given the valves location, each subproblem is a MFP whose dual is the minimum capacity cut provided in (6–7e). For each $s \in S$ there is a dual variable for each constraint of the MFP model.

$\omega^s_{i,\epsilon_{ij}}$, $\omega^s_{\epsilon_{ij},i} \geq 0, \forall (i,\epsilon_{ij}) \in F$ are the non negative variables associated to capacity constraints (5e) and (5f).

$\omega^s_{\epsilon_{ij},P} \geq 0, \forall \epsilon_{ij} \in \Psi$ are the non negative variables associated to capacity constraints (5g).

$\omega^s_{\sigma,i} \geq 0, \forall i \in \Gamma(\Sigma)$ are the non negative variables associated to capacity constraints (5h).

π^s_i $\forall i \in \Gamma \cup \Psi$ are the unconstrained node potential variables associated to flow balance constraints (5a) and (5b).

π^s_P and π^s_σ are the potential variables associated to the sink and the source flow balance constraints (5c) and (5d).

For each sector, an equivalent reformulation of the dual problem of (4-5i) is stated below.

$$min \sum_{(i,\epsilon_{ij})\in F} \Upsilon(1-\tau^s_{ij})(\omega^s_{i,\epsilon_{ij}} + \omega^s_{\epsilon_{ij},i}) + \sum_{(\epsilon_{ij},P)\in D} (\delta_{\epsilon_{ij}} \omega^s_{\epsilon_{ij},P}) \tag{6}$$

$$\pi^s_P - \pi^s_\sigma \geq 1, \tag{7a}$$

$$\pi^s_i - \pi^s_{\epsilon_{ij}} + \omega^s_{i,\epsilon_{ij}} \geq 0 \qquad \forall (i,\epsilon_{ij}) \in F, \tag{7b}$$

$$\pi^s_{\epsilon_{ij}} - \pi^s_i + \omega^s_{\epsilon_{ij},i} \geq 0 \qquad \forall (i,\epsilon_{ij}) \in F, \tag{7c}$$

$$\pi^s_{\epsilon_{ij}} - \pi^s_P + \omega^s_{\epsilon_{ij},P} \geq 0 \qquad \forall (\epsilon_{ij},P) \in D, \tag{7d}$$

$$\pi^s_\sigma - \pi^s_i + \omega^s_{\sigma,i} \geq 0 \qquad \forall (\sigma,i) \in T. \tag{7e}$$

The dual objective function coefficients of $\omega^s_{i,\epsilon_{ij}}$ and $\omega^s_{\epsilon_{ij},i}$ depend on $\tau^s_{i,\epsilon_{ij}}$. To linearize this non linear expression, for each sector we introduce two non negative variables $\mu^s_{i,\epsilon_{ij}}$ and $\mu^s_{\epsilon_{ij},i}$ for each edge $(i,\epsilon_{ij}) \in F$ and constraints (8a-8b), which realize the equivalence $\mu^s_{i,\epsilon_{ij}} = \omega^s_{i,\epsilon_{ij}}\tau^s_{ij}$.

$$\mu^s_{i,\epsilon_{ij}} \leq \omega^s_{i,\epsilon_{ij}} \qquad \forall (i,\epsilon_{ij}) \in F,\ \forall s \in S, \tag{8a}$$

$$\mu^s_{i,\epsilon_{ij}} \leq \Upsilon\tau^s_{ij} \qquad \forall (i,\epsilon_{ij}) \in F,\ \forall s \in S. \tag{8b}$$

$\mu^s_{i,\epsilon_{ij}}$ is no greater than $\omega^s_{i,\epsilon_{ij}}$ and it is forced to 0 when τ^s_{ij} is 0. Since its coefficient in the objective function to be minimized is $-\Upsilon < 0$, then $\mu^s_{i,\epsilon_{ij}}$ will be equal to $\omega^s_{i,\epsilon_{ij}}$ if $\tau^s_{ij} = 1$.

Now we can replace the objective function (4) of the inner problem for each $s \in S$ by constraint (9) that imposes the well known *max flow – min cut* optimality condition.

$$x^s_{P,\sigma} = \sum_{(i,\epsilon_{ij})\in F} \Big(\Upsilon(\omega^s_{i,\epsilon_{ij}} + \omega^s_{\epsilon_{ij},i}) - \Upsilon(\mu^s_{i,\epsilon_{ij}} + \mu^s_{\epsilon_{ij},i})\Big) + \sum_{(\epsilon_{ij},P)\in D} (\delta_{\epsilon_{ij}}\omega^s_{\epsilon_{ij},P}). \tag{9}$$

Finally, a single level MILP for BIVLP is given by:
$min_\tau \Delta$ *s. t.* (1a–1c), (2b), (3a–3d), and $\Delta \geq \Upsilon - x^s_{P,\sigma}$, (9), (8a-8b), (5a-5i), (7a–7e) $\forall s \in S$ which can be fed into any MILP solver and solved, as described in the next section.

3 Computational results and conclusions

We developed the MILP model of Section 2 for a real life WDS, the Apulian network serving Puglia, a region in the south of Italy, which was the case study also in [14], [2], and in [7]. A scheme is depicted in Figure 4: it is made of one reservoir, 23 junctions, and 33 pipes.

Running time is heavily affected by the maximum number of sectors. A rough estimation is the following: $s_{max} = cc + h_{max} - \sum_{\gamma\in\Gamma(\Sigma)}(deg(\gamma)-1)$, where $\Gamma(\Sigma)$ is the set of junctions γ adjacent to a reservoir $\sigma \in \Sigma$ (in Figure 4, $\Gamma(\Sigma) = \{1\}$); $deg(\gamma)$ is the degree of vertex γ; cc is the number of connected components obtained after edges $(\gamma, \epsilon_{\gamma j})$ have been removed,

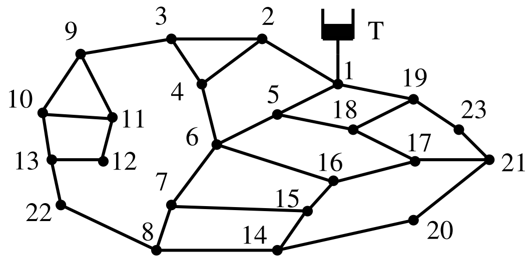


Figure 4 The Apulian network.

for each junction $\gamma \in \Gamma(\Sigma)$, except for the component made of σ, γ, and pipe (σ, γ). Any pipe with positive demand and incident on a junction in $\Gamma(\Sigma)$ must have a valve located near the junction to be isolated. Once such valves have been placed, yielding cc sectors, at most one more sector may result from the placement of one additional valve. This bound can be tight for very sparse networks, such as a tree, but real WDSs have several loops. A tighter estimate for s_{max} and a downsizing of the MILP model can be achieved by more sophisticated procedures exploiting the network topology of the instance, and are currently under investigation.

As in [2], h_{max} varies in $[5, \cdots, 14]$ and the time limit for each instance is 8000 seconds. In the Apulian network $cc = 1$, $\Gamma(\Sigma) = \{1\}$ and $deg(1) = 4$, then we have that $s_{max} = 1 + h_{max} - (4 - 1) = h_{max} - 2$, hence s_{max} varies in $[3, \cdots, 12]$.

The MILP solver is Gurobi Optimizer 4.6 [10], experiments were run on a *Intel dual core* architecture based on P8400 CPUs, 2.26 GHz, 4GB of RAM.

Figure 5 for each value of h_{max} reports the running time for the optimally solved instances, and the time the best feasible solution was found for the others. For such instances, labels report the gap between the best feasible MILP solution (z^{MILP}) and the optimal one provided in [2] (z^{CLP}). Note that for $h_{max} = 8, 9, 14$ we find the optimal solution within the time limit but we cannot certify it.

In conclusion, we provided a third exact solution approach to the BIVLP, based on MILP. A first implementation outperforms the ASP based approach and can solve to optimality

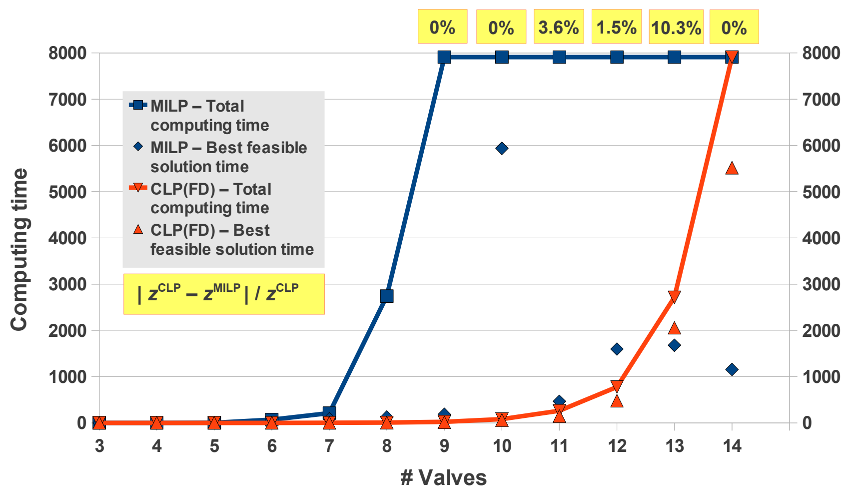


Figure 5 Computation time and solution gap values with different number of valves.

several instances of those solved by the CLP(FD) based approach in the same time limit, while provides good quality solutions to the others. Regarding the Pareto front generation, some iterations could be saved w.r.t. [2] where the number of used valves is fixed. Current research is devoted to improve the computational performance by: tuning the solver parameters which have been used at their default value; strengthening the GPP constraints taking advantage of the literature studies on its polyhedral structure [3, 6]; reducing symmetries in the model; tightening the estimate on s_{max} exploiting the WDS's graph topology; exploiting the knowledge of the exact solution with one more (one less) available valve in the computation of the Pareto optimal frontier. Furthermore, we aim to tackle different objective functions building on the MILP model here presented to formulate the feasible region of the ISD problem. For example, we plan to handle different probabilities of pipe failures and pipe dependent valve costs. Finally, hybrid approaches integrating MILP and CLP will be explored.

References

1 M. Bruglieri, P. Cappanera, A. Colorni, and M. Nonato. Modeling the gateway location problem for multicommodity flow rerouting. In J. Pahl et al., editor, *Network Optimization*, volume 6701 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2011.

2 M. Cattafi, M. Gavanelli, M. Nonato, S. Alvisi, and M. Franchini. Optimal placement of valves in a water distribution network with CLP(FD). *Theory and Practice of Logic Programming*, 11(4-5):731–747, 2011.

3 S. Chopra and M. R. Rao. The partition problem. *Mathematical Programming*, 59(1), 1993.

4 B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153:235–256, 2007. 10.1007/s10479-007-0176-2.

5 E. Creaco, M. Franchini, and S. Alvisi. Optimal placement of isolation valves in water distribution systems based on valve cost and weighted average demand shortfall. *Journal of Water Resources Planning and Management*, 24(15), 2010.

6 C. Ferreira, A. Martin, C. de Souza, R. Weismantel, and L. Wolsey. Formulations and valid inequalities for the node capacitated graph partitioning problem. *Mathematical Programming*, 74(3), 1996.

7 M. Gavanelli, M. Nonato, A. Peano, S. Alvisi, and M. Franchini. An ASP approach for the valves positioning optimization in a water distribution system. In F. Lisi, editor, *9th Italian Convention on Computational Logic (CILC 2012), Rome, Italy*, volume 857 of *CEUR Workshop Proceedings*, pages 134–148, 2012.

8 M. Gelfond. Answer sets. In *Handbook of Knowledge Representation*, chapter 7. Elsevier, 2007.

9 O. Giustolisi and D. A. Savić. Identification of segments and optimal isolation valve system design in water distribution networks. *Urban Water Journal*, 2010.

10 Gurobi Optimization, Inc. Gurobi optimizer reference manual. `http://www.gurobi.com`.

11 J. Jaffar and M. J. Maher. Constraint logic programming: A survey. *Journal of Logic Programming*, 19/20:503–581, 1994.

12 H. Jun and G. V. Loganathan. Valve-controlled segments in water distribution systems. *Journal of Water Resources Planning and Management*, 133(2), 2007.

13 J.-J. Kao and P.-H. Li. A segment-based optimization model for water pipeline replacement. *J. Am. Water Works Assoc.*, 99(7):83–95, 2007.

14 A. D. Nardo, M. D. Natale, G. Santonastaso, and S. Venticinque. Graph partitioning for automatic sectorization of a water distribution system. In D. Savic, Z. Kapelan, and D. Butler, editors, *Urban Water Management: Challenges and Opportunities*, volume 3, pages 841–846. Centre for Water Systems, University of Exeter, Exeter (UK), 2011.