IP Services Market: Modelling, Research and Reality

Piotr Arabas[1,2], Mariusz Kamola[1,2], and Krzysztof Malinowski[1,2]

[1] Warsaw University of Technology, Institute of Control and Computation Engineering, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
{P.Arabas, M.Kamola, K.Malinowski}@ia.pw.edu.pl
http://www.ia.pw.edu.pl

[2] Research and Academic Computer Network (NASK), ul. Wąwozowa 18, 02-796 Warsaw, Poland
{Piotr.Arabas, Mariusz.Kamola, Krzysztof.Malinowski}@nask.pl
http://www.nask.pl

Abstract. In the process of implementation and deployment of model-based tool for simulation, forecasting, decision support and optimisation, there discrepancies may emerge between the understanding of R&D team and the end user. Their origin is in the variety of optimisation problems a model is able to generate, depending on its parameters. A broad description of a market model for network services is given. Next, a case study follows that indicates a strategy for reduction of problems appearing while the product is handed over to the customer.

1 Introduction

The purpose of this paper is to present the outline of a model of network services market. Such model has been designed to be embedded in an optimisation solver to be then used for Network Services Provider (NSP) profit maximisation by setting service prices appropriately. Next, the authors focus on discrepancies that appeared between the model developer and the user regarding true nature of the simulation-based optimisation problem that emerged. Such discrepancies may result not from just imprecise problem definition but also from different notions of what actually can become the real obstacle for an optimisation task, as perceived by a development team with scientific background on one hand, and by the end user of the software on the other. Development and deployment of the simulation-optimisation module in the QOSIPS system will be the considered case study. It will be summarised with an account of how the resulting problems were solved in that particular case, and with a proposal of how to avoid such problems in the future.

QOSIPS is the acronym for "Quality of Service and Pricing Differentiation for IP Services" project whose main objectives were to develop innovative technologies for supporting QoS management, service differentiation and price setting of IP NSP's. The project was supposed to provide three major functionalities: accurate non-intrusive real-time QoS measurement of user's packets, service differentiation through traffic classification and use of QoS-oriented services, optimal composition and pricing of NSP services. One of the tasks within the Pricing Module workpackage was to develop a mathematical model of the IPN services (like VPN) offered to small and medium enterprises. After such a model is developed the model parameters are tuned, using either real life values (sales, utilisation etc.) or hypothetical scenarios, or just a rule of the thumb – depending on the available data. Once the model parameters are adjusted, this model can be used for prediction of market reaction to price changes, and even as a forecasting tool while introducing completely new products to the market. Embedding the market simulator in an optimisation routine allows the NSP to set the optimal product prices so that the profit (or other crucial parameters such as market share, or the mixture of them) is maximised.

The understanding of what the market model was expected to do was the same for all project partners, and resulted from careful investigation of the structure of the prevailing NSP tariffs. The modelling software closely followed its functional specification, was tested and accepted by the responsible parties in the project. Although the model details are commercially sensitive, some general outline will be given in Sect. 2. Next, the optimisation problem perception by the developer will be presented in Sect. 3, and confronted with that of the end user in Sect. 4. The conclusions are given in Sect. 5.

2 Market Model

The market modelling in QOSIPS was implemented to reflect the following phenomena:

- market segmentation,
- complexity of NSP offer (i.e. multi-component tariffs),
- usage-driven charging and cost schemes,
- customer activities (subscription, migration and churn),
- price-driven network utilisation,
- quality of service (QoS) degradation.

2.1 Market Segments

The market available to the NSP (and its competitors) can be divided in segments $S_1, S_2, ...$ Each segment is assumed to cluster potential customers having some distinct characteristics typical for them that may be identified. Market segmentation may be performed with respect to business type, budget level, usage profile and so on.

Market segmentation allows addressing ISP offer to the users, in particular to the market segments, basing upon common properties of the users grouped in a specific segment. Market segmentation is usually based on the customer features as follows: number of sites, number of roaming users, number of employees. The question arises, however, whether segmentation criteria presented above group potential customers properly. It seems that the number of sites or employees does not necessarily determine the traffic, usage or price sensitivity. Additional criteria that can give more appropriate market segmentation are the type of business, network/service usage pattern, type of application used and QoS requirements. A sample market segmentation is presented in Fig. 1.

Segment name	**Segmentation criteria**					
	Type of business		Usage pattern		Number of sites	
	Of low budget	Of high budget	Low data demand	High data demand	2	3
S_1	√		√		√	
S_2	√			√		√
S_3		√	√	√	√	
S_4		√	√	√		√

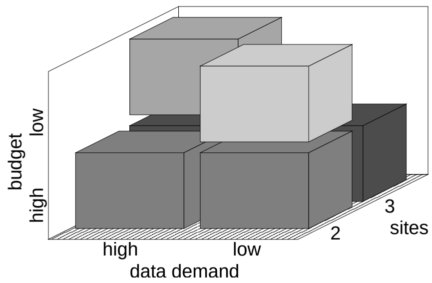

Fig. 1. Exemplary market segmentation presented by a table and graphically. Segment S_1 groups low-budget and low-data customers with two sites interconnected, and so on. The graphical representation splits the market represented by a large cube into smaller cubes of different colours that correspond segments — from the brightest (S_1) to the darkest (S_4). The three directions represent segmentation criteria. The two missing cubes represent the part of the market not being targeted

Additionally, each market segment can be characterized by its size, i.e. the total number of current and potential customers belonging to it.

2.2 Tariffs and Their Elements

NSP offer consists of hardware, data links, applications, services, support options etc. bundled in packages. A package with its contents attributed by prices is called tariff. Let us denote tariffs by $T_1, T_2, ...$ A tariff contains universal elements (components) that can stand for hardware, application, services – depending on the context. They can be arranged freely in a tree-like structure to represent variety of NSPs' products. Besides, a tariff contains extra parameters as its expiry date and competitor prices (i.e. any known prices in competitive tariffs that influence demand for that particular tariff).

Any tariff element is allowed to be compound of sub-elements, thus providing the way to construct the tree-like structure mentioned earlier. Regardless of that, a subscription for elements can be obligatory, optional or subject to "one-of-many" scheme. Each element is attributed by a standard set of fees/costs that are charged/incurred when a client subscribes, changes, renounces or simply accepts for another month the NSP services. Therefore, there are monthly fee, monthly cost, installation fee, installation cost and so on. To reflect NSP incomes and expenses dependant on tariff element utilization, pricing schemes are introduced in the model.

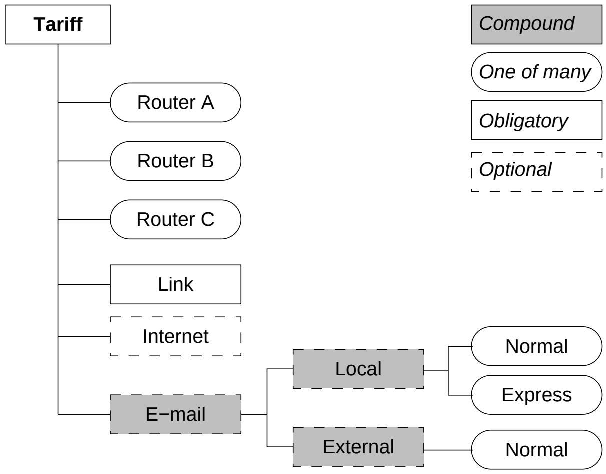

Fig. 2. An example of NSP offer represented as a tree of elements

Thanks to neutrality of elements, they can represent quantitative as well as qualitative diversity of the NSP offer. Let us look at Fig. 2 where elements serve equally well to describe hardware, data link, and services. The qualitative diversity (*Local/External* and *Normal/Express*) of mail is obtained in this case by extra branches hosting additional elements. The same could be done about e.g. the temporal service diversification (*Peak/Off-peak* hours, for example).

2.3 Usage-Based Schemes

Usage-based schemes are stepwise one-argument real-valued functions that are defined independently of other model entities. They are used to describe progressive pricing i.e. price that depends on some factor. Pricing schemes attribute tariff elements whose prices (and/or costs) are stepwise functions of that element utilization. Pricing schemes also serve in the calculation of compensation paid by NSP to the customer in case of not meeting QoS specification.

Let us denote usage based pricing schemes available in the system by $P_1, P_2, ...$ The definition of an exemplary pricing scheme and the graph of a corresponding stepwise function is presented in Fig. 3. This is, say, a charging scheme for data transfer. It defines that up to 6 MB (*Breakage level* column) the user is charged for every 2 MB (*Basic quantity*) by the amount of 2 (*Price per unit*). Having crossed the 6 MB boundary, the charging is as defined in the next line of the table.

2.4 Customer Activities

Tariffs can be offered to various segments. It is on the intersection of segment S_i and tariff T_j where customer behaviour can be measured because it takes form

Breakage level	Price per unit	Basic quantity
6	2	2
n/a	1	3

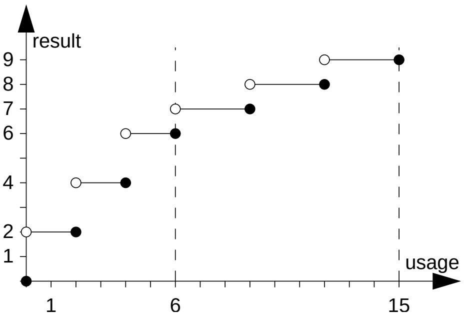

Fig. 3. Table defining a pricing scheme (left) and the corresponding graph (right)

of concrete numbers of new, migrating and churning customers, and of their distribution across the tariff elements.

Table 1. Sample assignment of tariffs to market segments

Market segments	T_1	T_2	T_3	$\cdots$	T_n
S_1	$\surd$	$\surd$			
S_2		$\surd$	$\surd$		
S_3		$\surd$	$\surd$		
$\vdots$	$\vdots$			$\ddots$	$\vdots$
S_m					$\surd$

A single tariff may be addressed to multiple market segments, as it is presented in Table 1. With each tariff/segment intersection there is associated a state variable $x_{i,j}$ denoting the number of customers of tariff T_j in segment S_i. This number will change in time as new customers arrive, attracted by product prices, while the existing ones may stay, migrate or churn. The value of $x_{i,j}$ in month $t+1$ is determined by the following formula:

$$x_{x,j}(t+1) = x_{i,j}(t) + x_{i,j}^{new}(t+1) + x_{i,j}^{mig}(t+1) - x_{i,j}^{churn}(t+1) \ , \qquad (1)$$

where:

- $x_{i,j}(t)$ is the number of customers in the preceding month,
- $x_{i,j}^{new}(t+1)$ is the number of new customers,
- $x_{i,j}^{mig}(t+1)$ is the balance of migrating customers from and to other tariffs in the same market segment,
- $x_{i,j}^{churn}(t+1)$ is number of churning customers.

The number of new, migrating and churning customers are calculated using each time the widely known Cobb-Douglas formula (cf. [4, 2]):

$$x(\mathbf{p}) = \alpha \prod_{k=1}^{\dim \mathbf{p}} p_k^{\beta_k} \ , \tag{2}$$

where $\mathbf{p} = (p_1, \ldots, p_k)$ is the vector of selected prices (own and competitors) or QoS metrics experienced in the previous month, and α, β_k are the function parameters: the scaling factor and sensitivities.

The distribution of $x_{i,j}$ across tariff elements is assumed constant (fixed). Such modelling simplification was introduced because of the amount of data too scarce to initialise many submodels that would be needed in case of exact modelling the number of customers for every sub-element.

2.5 Resource Utilisation

Utilization can be modelled for every tariff element, at the level of tariff/segment intersection. For this the purpose formula (2) is used but this time without previous QoS metrics as arguments. The meaning of the word "usage" depends on what actually the element is: e.g. for a data link it can be the total online time over a month, for an e-mail it can be the total volume of all mail transmitted etc.

2.6 QoS Degradation

Network traffic, represented by tariff elements utilization, may cause that the quality parameters of some services, guaranteed by provider in the contract, are not met. QoS guarantees depend on the nature of a particular application: for telnet it can be "transmission delay less than 10 millisecond", for video on demand it can be "packet loss rate less than 2%", and alike. In QOSIPS, one can associate with an element a number of QoS clauses like those above. If QoS requirements are not met, then a network provider must pay penalties dependent on the rate QoS degraded.

Let the QoS metric be the percent of data (or time) for which the QoS guarantee was not met. Therefore, a QoS metric can take values between 0 and 1, and obviously is determined by NSP infrastructure, customers' habits and locations. As there is no simple, precise and universal QoS modelling mechanism, rough formula has been put in the market model:

$$q(\mathbf{x}, \mathbf{u}) = \min\left(1, \max\left(0, a + b \sum_{k=1}^{\dim \mathbf{u}} x_k u_k\right)\right) \ , \tag{3}$$

which is but a linear function whose values are limited by 1 from above and by 0 from below. The function argument is what we have called *aggregated traffic*, i.e. the sum of products of indicated element utilizations u_k and the corresponding number of users x_k for that element. This simple function has the advantage of having only two coefficients (a and b) and it is easy choose their values.

To compute the compensation for not meeting the QoS parameters, q is simply fed into one of progressive pricing schemes – the bigger QoS offence, the more money back to the customer. The employed stepwise functions allow for accommodating of the payback scheme to the individual nature of a subscriber. Adaptation of the compensations to a personal preferences of a user is an analogue for more complex metods of operational traffic control, e.g those based on the notion of utility function. For a discussion on the relations between pricing on the marketing and network operational level, see [6, 7].

3 Development and Academic Model

In the process of the Mathematical Library Component development (MLC – the piece of the QOSIPS system responsible for forecasting and price optimisation), the R&D team produced many test problems to check the module against the existence of errors or to verify its compliance with the specification. For the purpose of this paper another simple market model has been created to demonstrate the crucial features of the MLC. It is composed of one tariff only, instead of the usual multitude of options (e.g. distribution across elements, tariff expiry dates, competitor prices); it models just the monthly volume of new and churning subscribers, the average network usage per subscriber, and the compensation paid when the NSP fails to provide the subscribers with appropriate quality of service.

Now it is time to mention the question of profit calculation that has not been touched while describing the model. The profit for each tariff element is calculated intuitively, by subtraction of monthly costs and QoS compensations from the income. The overall profit is the total of tariff element profits over a given period (6 months, on usual). Simple as it may seem, the profit calculation involves considerably difficult issues of putting some uncertainty in the model. That uncertainty initially did not exist, extinguished by modeling only means (of the number of customers, of the usage etc.) but it played an important role in calculations. Refinement of algorithms applied in MLC is still a subject under research.

In the considered simplified model there are only two prices that influence customer behaviour and determine the NSP profit. They are: the monthly fee and the price for transmitting a unit of data. There is also a credit for data, contained within the monthly fee. Usually, those two price variables are subject to additional constraints, i.e. they must lie within some interval determined either by the validity of the model or by market constraints (e.g. NSP own prices cannot differ too much from the competitive offer). Let us consider the first of them – the monthly fee. It is felt that normally the price that results in the highest profit must be neither too low nor too high. When low, there are many new customers and few leavers, but their number cannot compensate for losses due to the low price level. When the price is high, the customers will soon desert to the competitors. Therefore, the optimal price must lie somewhere in the middle. The negated value of the NSP profit due to the recurrent fee in the academic model is drawn in Fig. 4a). From now on all the figures and tables will represent negated values of profit in order to convert the profit maximisation task into a standard objective function minimisation problem.

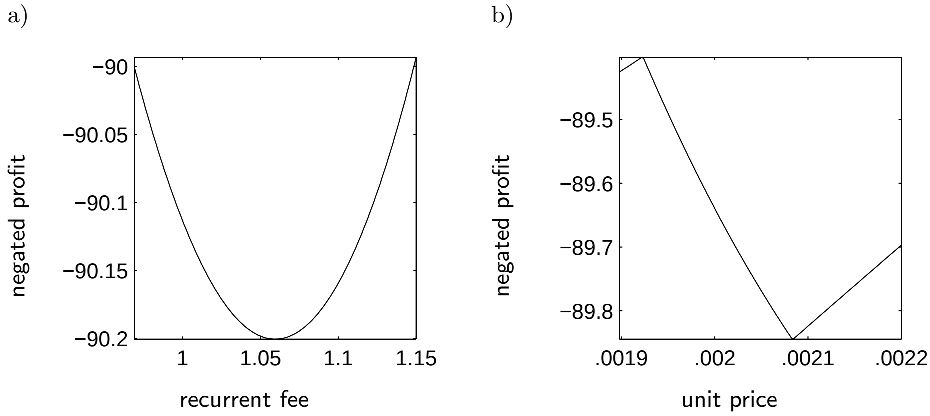

Fig. 4. a) Profit against the recurrent fee (for the unit price of 0.0022) **b)** Profit against the unit price (for the recurrent fee of 1.2)

There is also another price variable – the price of data unit. Like in the case of a monthly fee, raising the unit price discourages subscribers from increasing the volume of traffic beyond that covered by the fixed fee. Alternatively, lowering prices induces traffic volumes beyond what the network can transfer, and the NSP ends up paying high compensation to the users. Here again the optimum price lies somewhere in the middle, as shown in Fig. 4b). The contour plot of the performance function of both price variables is shown in Fig. 5.

In Fig. 5 one can easily observe irregularities in the level contours caused by activation of QoS paybacks that create additional, local optima. The development team expected them to be potentially the main cause of problems for the end user – especially when the starting point is away from the optimum. CRS2 and COMPLEX routines were employed for the task of optimisation, with modifications necessary to support extra constraints, whose description lies outside the scope of this paper. Also a proprietary solver developed by the QOSIPS project participants has been tested against this simple optimisation task.

- CRS2 is a global direct search method, described first in [3]. It maintains a set of points (randomly chosen at the beginning) and, by successive elimination of the worst ones and reflection of the best ones, slowly but steadily approaches the minimum. It performs best while far from the solution; in its proximity (i.e. where stronger assumptions as to the function shape can be made) it should be replaced by a more effective routine.

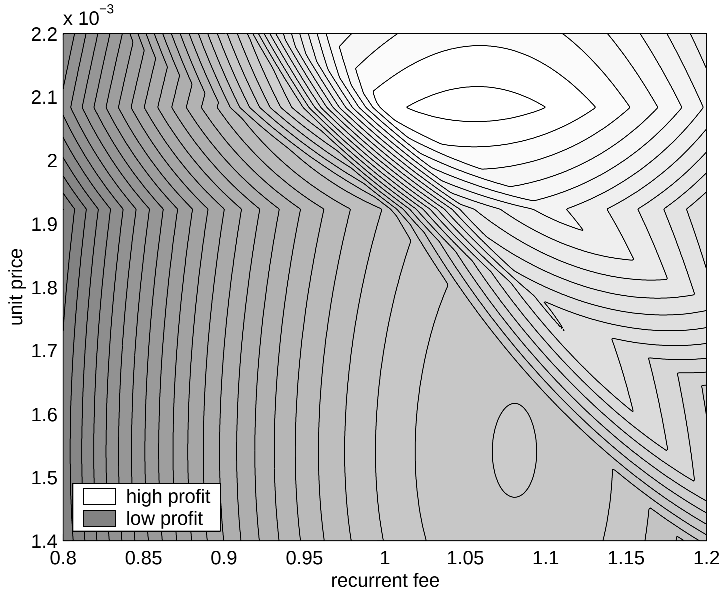

Fig. 5. Profit against the recurrent fee and the unit price

- COMPLEX, being similar in its workings to CRS2 (point pool, reflections), differs in that it has a smaller number of maintained points and converges less surely to global optimum. Instead, it supports complicated and implicitly defined constraints and performs better than CRS2 near the optimum. For the method details, see [1].
- Proprietary routine performs at each step the linearisation of the objective function and constraints, and solves the linear subproblem. Then, based on the past linear subproblems solutions, it computes the next point where the linear subproblem is solved. The routine is the result of QOSIPS participant's long-term experience in the field of optimisation: it incorporates amendments that improve the efficiency, prevent preliminary termination etc.

A series of price optimisations has been run for all the above solvers, from various starting points. The starting points are shown in Fig. 6a) and the optimum prices found by CRS2, COMPLEX and the proprietary solver are indicated in Figs. 6b-d), respectively. Axes descriptions have been omitted for clarity – they are the same as in Fig. 5. The best result for each solver has been put in Table 2.

The CRS2 and COMPLEX routines perform best – they yield optimal or almost optimal value of the performance function. They both converge globally, but CRS2 does not converge exactly to the minimum as the COMPLEX does – which corresponds to their general characteristics. On the other hand, the proprietary method solutions depend on the starting point – the method sticks sometimes at the local minimum at $[1.0731, 0.00154]$ yielding the profit value

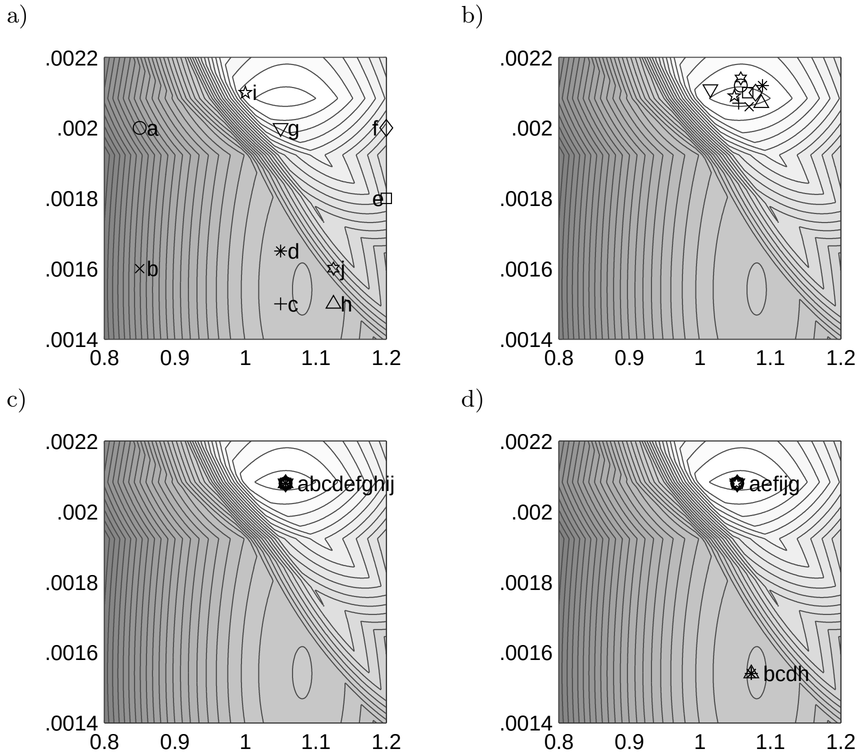

Fig. 6. a) Optimisation starting points (common for all tested solvers) **b)** Solutions found by CRS for the corresponding starting points **c)** Solutions found by COMPLEX for the corresponding starting points **d)** Solutions found by the proprietary routine for the corresponding starting points

88.85. The method does very well with gradient estimations even at the surface irregularities, but due to its local character it was not further considered for practical applications by the R&D team.

4 End User Model

The Academic team's perception of the potential problems in simulation and optimisation phases was not shared by the end user – the NSP and the team responsible for model tuning, deployment and field testing. Not having enough real life data concerning sales and network utilisation, they have constructed a model based on their long-term experience in product pricing in other fields. There, the following possible difficulties disappeared:

- **sharp edges of the performance function contours** caused by rapidly growing QoS paybacks – because the network QoS degradation model saturated slowly, thus facilitating gradient estimation;

Table 2. Academic example – the best optimisation results for each tested solver

Routine	Recurrent fee	Unit price	Profit
CRS2	1.0494	0.00209	90.35
COMPLEX	1.0568	0.00208	90.37
proprietary	1.0528	0.00208	90.35

– **multiple optima** – because model parameters made the performance function unimodal.

Two simple two-dimensional sections of the performance function surface are shown in Fig. 7a) and Fig. 7b). In this example there are 22 price variables being subject to the optimisation. The sections demonstrate that the function grows steadily, making the optimisation problem like as in linear programming, as the optima are always located in a corner of the feasible polyhedron set – the optimisation domain. In such circumstances the former winning routines like CRS2 or COMPLEX may promptly get stuck in a corner remote from the solution. On the other hand, the proprietary algorithm, once designed to handle similar pricing problems, performs well.

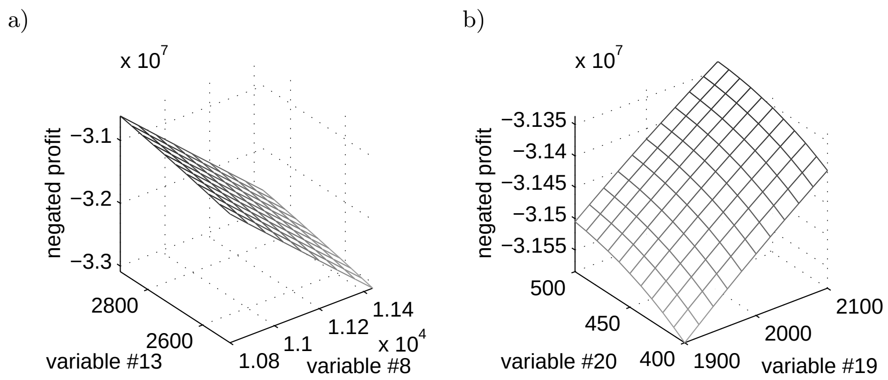

Fig. 7. a) Profit against variables #8 and #13 **b)** Profit against variables #19 and #20

There was another reason for CRS2 and COMPLEX failures – the way the linear equality constraints were defined by the user. The user was accustomed to introduce them by defining pairs of nonequality constraints, which was his habit for former products. The result was a zero-measure search domain with an almost empty interior, unsupportable by the direct search methods. The proprietary algorithm, apt for this type of constraint definition, worked undisturbed.

5 Conclusions

As the case study shows, despite the model being defined and working properly, most of its behaviour towards the optimisation routine is defined by the model parameters. Correct (i.e. based on the realistic data) values of those parameters are usually not known to the R&D team that implements the model and plugs it into an optimisation solver. Moreover, they may not even be known for the end user, until the real data arrive.

The suggested strategy for R&D teams is then as follows:

1. Assess qualitatively what kinds of optimisation problems a given model may generate.
2. Consider a moderate set of well-tested optimisation algorithms appropriate for those problems.
3. Design the simulation/decision support module so that the user may choose which optimisation routine to utilise, and to use a solution found by one optimisation routine as a starting point for another.
4. Acquaint the user with the capabilities of the module, to change old habits that may decrease the effectiveness of the model and the optimisation solver.

6 Acknowledgments

This research was carried out within the IST Project 1999-20003 “Quality of Service and Pricing Differentiation for IP Services” (QOSIPS) of the 5th Framework Programme of the European Union.

The authors thank Xiao-Jun Zeng and Vicky Tsamadia from KSSG Ltd. for their fruitful cooperation in research, testing and in the application of the described software.

References

1. Box, M.J.: A new method of constrained optimisation and a comparison with other methods. The Computer Journal **8**:1 (1965) 42–52
2. Lilien, G.L., Kotler, P., Moorthy, K.S.: Marketing Models. Prentice Hall, Upper Saddle River (1992)
3. Price, W.L.: Global optimization algorithms for a CAD workstation. Journal of Optimisation Theory and Applications **55**:1 (1987) 133–146
4. Simon, H.: Price Management. Elsevier Science Publishers, Amsterdam (1989)
5. QOSIPS Deliverable 3.2.1: Research report on how pricing model, learning and optimization are to be implemented to satisfy requirements. (2001)
6. QOSIPS Deliverable 6.7.2: Pricing for IP networks and Services. (2001)
7. Arabas, P., Kamola, M., Malinowski, K., Małowidzki, M.: Pricing for IP Network and Services. Information Knowledge Systems Management **3**:2 (2003) (to appear)