



**HAL**  
open science

# On Combining Regression Analysis and Constraint Programming

Carmen Gervet, S Galichet

► **To cite this version:**

Carmen Gervet, S Galichet. On Combining Regression Analysis and Constraint Programming. IPMU, 2014, Montpellier, France. hal-01800672

**HAL Id: hal-01800672**

**<https://hal.science/hal-01800672>**

Submitted on 27 May 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# On Combining Regression Analysis and Constraint Programming\*

C. Gervet and S. Galichet

LISTIC, Laboratoire d'Informatique, Systems, Traitement de l'Information et de la  
Connaissance - Universit de Savoie, BP 80439  
74944 Annecy-Le-Vieux Cedex France  
gervetec,sylvie.galichet@univ-savoie.fr

**Abstract.** Uncertain data due to imprecise measurements is commonly specified as bounded interval parameters in a constraint problem. For tractability reasons, existing approaches assume independence of the parameters. This assumption is safe, but can lead to large solution spaces, and a loss of the problem structure. In this paper we propose to combine the strengths of two frameworks to tackle parameter dependency effectively, namely constraint programming and regression analysis. Our methodology is an iterative process. The core intuitive idea is to account for data dependency by solving a set of constraint models such that each model uses data parameter instances that satisfy the dependency constraints. Then we apply a regression between the parameter instances and the corresponding solutions found to yield a possible relationship function. Our findings show that this methodology exploits the strengths of both paradigms effectively, and provides valuable insights to the decision maker by accounting for parameter dependencies.

**Key words:** Data uncertainty, Constraint Programming, regression analysis

## 1 Introduction

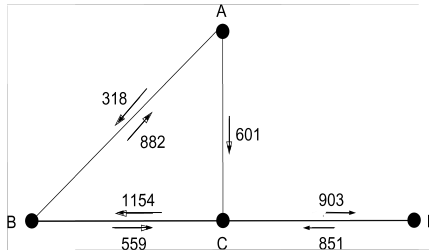
Data uncertainty due to imprecise measurements or incomplete knowledge is ubiquitous in many real world applications, network design, renewable energy investment planning, and inventory management, to name a few. Regression analysis is one of the most widely used statistical techniques to model and represent the relationship among variables to describe or predict phenomena for a given context. Recently, models derived from fuzzy regression have been defined to represent incomplete and imprecise measurements in a contextual manner, using intervals [2]. Such models apply to problems in finance or complex systems analysis in engineering whereby a relationship between crisp or fuzzy measurements is sought.

Constraint Programming (CP) on the other hand, is a powerful paradigm used to solve decision and optimization problems in areas as diverse as planning, scheduling, routing. The CP paradigm models a decision problem using

---

\* This research was partly support by the Marie Curie CIG grant, FP7-332683

constraints to express the relations between variables, and propagates any information gained from a constraint onto other constraints. When data imprecision is present, forms of uncertainty modeling have been embedded into constraint models using bounded intervals to represent such parameters. For instance if we model traffic in a network, the flow over links between two routers cannot be measured with precision as it depends on the collected traffic volume data at each router which cannot be synchronized. As a result, the data information obtained is erroneous. We use the sigcomm4 network given in Fig. 1 as a running example taken from [8]. On link  $A \rightarrow C$ , for example, the traffic volume might measure as 565 at A and as 637 at C, whereas the true value, equal at both nodes, is presumably somewhere in between. Thus, the flow between  $A$  and  $C$  will be specified by  $565.0..637.0$ . The mean values are shown in the figure. Such models are used to find the flow matrix between any pair of routers. Commonly flow distribution is considered between 30 and 70 percent. Thus on the link  $A \rightarrow C$  the flow constraint is specified by:  $0.3..0.7 * F_{AC} + 0.3..0.7 * F_{BC} + 0.3..0.7 * F_{AB} + 0.3..0.7 * F_{AD} = 565.0..637.0$ . A variable  $F_{AD}$  denotes the traffic (unknown) from router  $A$  to router  $D$ .



**Fig. 1.** Sigcomm4 network topology and mean values for link traffic

Traditional models either omit any routing uncertainty for tractability reasons, and consider solely the shortest path routing or embed the uncertain parameters but with no dependency relationships. Values for the flow variables are derived by computing bounded intervals, which are safe enclosing of all possible solutions. Such intervals enclose the solution set without relating to the various instances of the parameters. For instance, the traffic between  $A$  and  $C$  can also pass through the link  $A \rightarrow B$ . Thus the flow constraint on this link also contains  $0.3..0.7 * F_{AC}$ . However, the parameter constraint stating that the sum of the coefficients of the traffic  $F_{AC}$  in both constraints should be equal to 1 should also be present. Assuming independence of the parameters for tractability reasons, leads to safe computations, but at the potential cost of a very large solution set, even if no solution actually holds. The problem structure is lost. Also, there is not insight as to how the potential solutions evolve given instances of the data.

The question remains as to how can this information be embedded in a constraint model that would remain tractable. To our knowledge this issue has not been addressed in the general case. It is the purpose of this work.

In this paper we propose a new methodology and efficient process to account for data dependency constraints in decision problems. We aim to more closely model the actual problem structure, refine the solutions produced and add accuracy to the decision making process. We use regression analysis to show the relationship among various instances of the uncertain data parameters and the solutions produced. The basic process is to extract the parameter constraints from the model, solve them to obtain tuple solutions over the parameters. Then we run simulations on the constraint models using the parameter tuples as data instances that embed the dependencies. In the example above this would imply for the two constraints given, that if one parameter takes the value 0.3, the other one would take the value 0.7. A set of constraint models can thus be solved efficiently, matching a tuple of consistent parameters to a solution, to determine whether there are solutions once dependencies are taken into account or not. Finally, we run a regression analysis between the parameter values and solutions produced to determine the regression function, i.e. see how potential solutions relate to parameter variations.

The paper is structured as follows. Section 2 summarizes the related work and gives a small illustrative case study. Section 3 describes the approach and algorithms, and Section 4 gives an application study. A conclusion is given in Section 5.

## 2 Background and case study

The fields of regression analysis and constraint programming are both well established in computer science. While we identified both fields as complementary, there has been little attempt to integrate them together to the best of our knowledge. The reason is, we believe, that technology experts tackle the challenges in each research area separately. However, each field has today reached a level of maturity shown by the dissemination in academic and industrial works, and their integration would bring new research insights and a novel angle in tackling real-world optimization problems with measurement uncertainty. There has been some research in Constraint Programming (CP) to account for data uncertainty, and similarly there has been some research in regression modeling to use optimization techniques. We give an account of the state of the art against our main objective to integrate both paradigms.

CP is a paradigm within Artificial Intelligence that proved effective and successful to model and solve difficult combinatorial search and optimization problems from planning and resource management domains [9]. Basically it models a given problem as a Constraint Satisfaction Problem (CSP), which means: a set of variables, the unknowns for which we seek a value, the range of values allowed for each variable, and a set of constraints which define restrictions over the variables. Constraint solving techniques have been primarily drawn from Artificial Intelligence (constraint propagation and search), and more recently Operations Research (graph algorithms, Linear Programming). A solution to a constraint model is a complete consistent assignment of a value to each decision variable.

In the past 15 years, the growing success of constraint programming technology to tackle real-world combinatorial search problems, has also raised the question of its limitations to reason with and about uncertain data, due to incomplete or imprecise measurements, (e.g. energy trading, oil platform supply, scheduling). In the past 10 years, the generic CSP formalism has been extended to account for forms of uncertainty: e.g. numerical, mixed, quantified, fuzzy, uncertain CSP and CDF-interval CSPs [3]. The fuzzy and mixed CSP [7] coined the concept of parameters, as uncontrollable variables, meaning they can take a set of values, but their domain is not meant to be reduced to one value during problem solving (unlike decision variables). Constraints over parameters, or uncontrollable variables, can be expressed and thus some form of data dependency modeled. However, there is a strong focus on discrete data, and the consistency techniques used are not always effective to tackle large scale or optimization problems.

Frameworks such as *numerical, uncertain, or CDF-interval* CSPs, extend the classical CSP to approximate and reason with continuous uncertain data represented by intervals; see the real constant type in Numerica [11] or the bounded real type in ECLIPSe [4]. Our previous work introduced the *uncertain and CDF-interval* CSP [12, 10]. The goal was then to derive efficient techniques to compute reliable solution sets that ensure that each possible solution corresponds to at least one realization of the data. In this sense they compute an enclosure of the set of solutions. Even though we identified the issue of having a large solution set, the means to relate different solutions to instances of the uncertain data parameters and their dependencies were not thought of. On the other hand, in the field of regression analysis, the main challenges have been in the definition of optimization functions to build a relevant regression model, and the techniques to do so efficiently. Regression analysis evaluates the functional relationship, often of a linear form, between input and output parameters in a given environment. Here we are interested in using regression to seek a possible relation between uncertain constrained parameters in a constraint problem, e.g. distribution of traffic among two routers on several routes and the solutions computed according to the parameter instances. We note also that methods such as sensitivity analysis in Operations Research allow to analyze how solutions evolve relative parameter changes. However, such models assume independence of the parameters. In the following case study we show how our approach can establish relationships with the solutions and uncertain parameters while accounting for dependencies.

*Case study* We present a small case study to give the intuition of our approach. The core element is to go around the solving of a constraint optimization problem with uncertain parameter constraints by first solving the parameter constraints alone. This way we handle uncertainty in an efficient and tractable manner. We then substitute solution tuples of these parameters to solve a set of constraint optimization problems (now without parameter constraints). Finally to provide insight to the solutions to the uncertain constraint problem, we run a regression between the solutions produced and the corresponding parameter tuple instances.

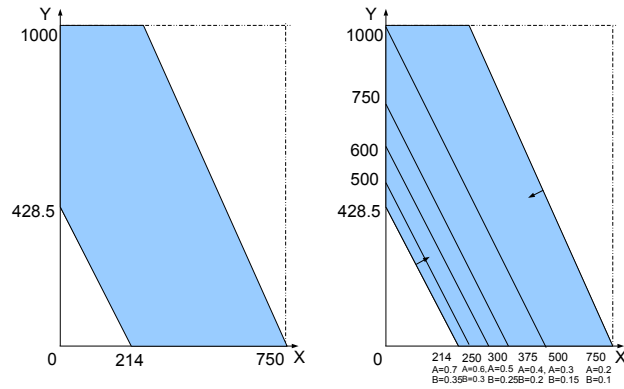
Consider the following fictitious constraint between two unknown positive variables,  $X$  and  $Y$  ranging in  $0.0..1000.0$ , with uncertain data parameters  $A, B$  taking their values in the real interval  $[0.1..0.7]$ :

$$A * X + B * Y = 150$$

The objective is to compute values for  $X$  and  $Y$  in the presence of uncertain parameters ( $A, B$ ). Without any parameter dependency a constraint solver based on interval propagation techniques with bounded coefficients, derives the ranges  $[0.0..1000.0]$  for both variables  $X$  and  $Y$  [4]. Let us add to the model a parameter constraint over the uncertain parameters  $A$  and  $B$ :  $A = 2 * B$ . Without adding this parameter constraint to the model, since it is not handled by the solver, we can manually refine the bounds of the uncertain parameters in the constraint model such that the parameter constraint holds over the bounds, thus accounting partially for the dependency. We obtain the constraint system:

$$[0.2..0.7] * X + [0.1..0.35] * Y = 150$$

The solution returned to the user is a solution space:  $X \in [0.0..750.0], Y \in [0.0..1000.0]$ . The actual polyhedron describing the solution space is depicted in Fig. 2.



**Fig. 2.** Left: Solution space. Tight and certain bounds for the decision variables:  $[0, 750]$   $[0, 1000]$ . Right: Solutions vectors of problem instances with consistent parameter solutions

We now give the intuition of our approach. The idea is to first solve the parameter dependency constraints alone to obtain solution tuples, not intervals. To do so we use a traditional branch and bound algorithm. We obtain a set of tuples for  $A$  and  $B$  such that for each tuple the constraint  $A = 2 * B$  holds. The idea is to have a well distributed sample of solutions for this parameter constraint.

We obtain a set of tuples that satisfy the parameter constraint, in this case for instance  $(0.2, 0.1)$ ,  $(0.3, 0.15)$ ,  $(0.4, 0.2)$ ,  $(0.5, 0.25)$ ,  $(0.6, 0.3)$ ,  $(0.7, 0.35)$ . We then substitute each tuple in the uncertain constraint model rendering it a standard constraint problem, and solve each instance. We record the solution matching each parameter instance. The issue now is that we have a set of solutions for each tuple of parameters. There is no indication how the solutions are related and evolve. The idea is to apply a regression analysis between both. The regression function obtained includes the solution bounds obtained by the standard approach, but mainly shows the trends between the data parameters and the solutions. In this small example we can visualize how the solution evolves with the data, see Fig. 2 on the right. In the case of much larger data sets, a tool like Matlab can be used to compute the regression function and display the outcome. The algorithm and complexity analysis are given in the next section.

### 3 Process and algorithms

Our methodology is a three-steps iterative process: 1) Extract the uncertain parameter constraints from the uncertain optimization problem and run branch and bound to produce a set of tuple solutions, 2) solve a sequence of standard constraint optimization problems where the tuples are being substituted to the uncertain parameters. This is a simulation process that produces, if it exists, one solution per tuple instance. And finally, 3) run a regression analysis on the parameter instances and their respective solution, to identify the relationship function showing how the solution evolves with consistent parameter constraints. The overall algorithmic process is given in Fig. 3, where the outcomes at each step are highlighted in italic bold. A constraint satisfaction and optimisation problem, or CSOP, is a constraint satisfaction problem (CSP) that seeks complete and consistent instantiations optimizing a cost function. We use the notion of uncertain CSOP, or UCSOP first introduced in [12]. It extends a classical CSOP with uncertain parameters.

#### 3.1 Uncertain CSOP and uncertain parameter constraints

Recall that a CSOP is commonly specified as a tuple  $(\mathcal{X}, \mathcal{D}, \mathcal{C}, f)$ , where  $\mathcal{X}$  is a finite set of variables,  $\mathcal{D}$  is the set of corresponding domains,  $\mathcal{C} = \{c_1, \dots, c_m\}$  is a finite set of constraints, and  $f$  is the objective function (min or max of a given expression over a subset of the variables).

**Definition 1 (UCSOP).** *An uncertain constraint satisfaction and optimisation problem is a classical CSOP in which some of the constraints may be uncertain, and is specified by the tuple  $(\mathcal{X}, \mathcal{D}, \mathcal{C}_{\mathcal{X}}, \Lambda, \mathcal{U}, f)$ . The finite set of parameters is denoted by  $\Lambda$ , and the set of ranges for the parameters by  $\mathcal{U}$ . A solution to a UCSOP is a solution space enclosing safely the set of possible solutions.*

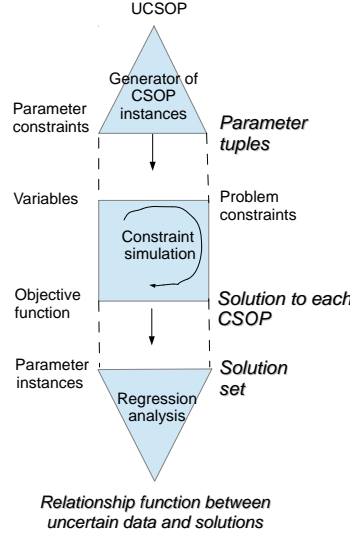


Fig. 3. Process

*Example 1.* Let  $X_1 \in D_1$  and  $X_2 \in D_2$  both have domains  $D_1 = D_2 = [1.0..7.0]$ . Let  $\lambda_1$  and  $\lambda_2$  be parameters with uncertainty sets  $U_1 = [2.0..4.0]$  and  $U_2 = [1.0..6.0]$  respectively. Consider three constraints:

$$C_1 : X_1 > \lambda_1, \quad C_2 : X_1 = X_2 + \lambda_2, \quad C_3 : X_2 > 2, \quad C_4 : \lambda_2 = \lambda_1 + 3$$

and the objective function to maximize  $f(X_1, X_2) = X_1 + X_2$ . Writing  $\mathcal{X} = \{X_1, X_2\}$ ,  $\mathcal{D} = \{D_1, D_2\}$ ,  $\Lambda = \{\lambda_1, \lambda_2\}$ ,  $\mathcal{U} = \{U_1, U_2\}$ , and  $\mathcal{C}_{\mathcal{X}} = \{C_1, C_2, C_3\}$ , then  $(\mathcal{X}, \mathcal{D}, \mathcal{C}_{\mathcal{X}}, \Lambda, \mathcal{U}, f)$  is a UCSOP. Note that  $C_3$  is a certain constraint;  $C_1$  and  $C_2$  are both uncertain constraints because they contain uncertain parameters.  $\mathcal{C}_{\Lambda} = \{C_4\}$  is the set of parameter constraints.

### 3.2 Constraint simulation

We now account for the parameters constraints by transforming the UCSOP into a set of tractable CSOPs instances. More formally, consider a UCSOP  $(\mathcal{X}, \mathcal{D}, \mathcal{C}_{\mathcal{X}} \cup \mathcal{C}_{\Lambda}, \Lambda, \mathcal{U}, f)$ .

**Definition 2 (instance of UCSOP).** We denote  $n$  the number of variables,  $m$  the number of uncertain parameters,  $p$  the number of parameter constraints, and  $inst(\mathcal{U}_i)$  a value within the range of an uncertainty set. An instance of an UCSOP is a certain CSOP  $(\mathcal{X}, \mathcal{D}, \mathcal{C}_{\mathcal{X}})$  such that for each uncertain constraint  $C_i(X_1..X_m, \lambda_1, ..\lambda_m)$ , we have  $\lambda_j = inst(\mathcal{U}_j)$ , such that  $\forall k \in \{1, .., p\}$ , the parameter constraint  $C_k(\lambda_1, ..\lambda_m)$  is satisfied.



*Example 2.* Continuing example 1, the UCSOP has two possible instances such that the parameter constraint  $\lambda_2 = \lambda_1 + 3$  holds, given that  $\lambda_1 \in \mathcal{U}_1, \lambda_2 \in \mathcal{U}_2$ . The valid tuples  $(\lambda_1, \lambda_2)$  are  $(2, 5)$ , and  $(3, 6)$ . The CSOP instances are:

$$C_1 : X_1 > 2, C_2 : X_1 = X_2 + 5, C_3 : X_2 > 2$$

and

$$C_1 : X_1 > 3, C_2 : X_1 = X_2 + 6, C_3 : X_2 > 2$$

with the same objective function to maximize  $f = X_1 + X_2$ .

The generator of CSOP instances extracts the parameter constraints, polynomial in the number of constraints in the worst case, then produces a set of parameter tuples that satisfy the parameter constraints. We can use a branch and bound search on the parameter constraints of the UCSOP. The constraint simulation then substitutes the tuple solutions onto the original UCSOP to search for a solution to each optimization problem, that is each CSOP. This is polynomial in the complexity of the UCSOP. The process is depicted in Algorithm 1.

---

**Algorithm 1:** Generate and solve CSOPs from one UCSOP

---

**Input:** A UCSOP  $(\mathcal{X}, \mathcal{D}, \mathcal{C}_X \cup \mathcal{C}_A, A, \mathcal{U}, f)$

**Output:** Solutions to the CSOPs

```

1 SolsTuples  $\leftarrow \emptyset$ 
2 extract( $\mathcal{C}_A$ )
3 Tuples  $\leftarrow solveBB(A, \mathcal{U}, \mathcal{C}_A)$ 
4 for  $T_i \in Tuples$  do
5   substitute  $A$  with  $T_i$  in  $(\mathcal{X}, \mathcal{D}, \mathcal{C}_X, A, f)$ 
6    $S_i \leftarrow solveOpt(\mathcal{X}, \mathcal{D}, \mathcal{C}_X, T_i, f)$ 
7   SolsTuples  $\leftarrow SolsTuples \cup \{(S_i, T_i)\}$ 
8 return SampleSols

```

---

## 4 Application

We illustrate the benefits of our approach by solving an uncertain constraint optimization problem, the traffic matrix estimation for the sigcomm4 problem, given in Fig. 1. The topology and data values can be found in [8, 12]. Given traffic measurements over each network link, and the traffic entering and leaving the network at the routers, we search the actual flow routed between every pair of routers. To find out how much traffic is exchanged between every pair of routers, we model the problem as an uncertain optimization problem that seeks the min and max flow between routers such that the traffic link and traffic conservation constraints hold. The traffic link constraints state that the sum of traffic using the link is equal to the measured flow. The traffic conservation constraints, two

per router, state that the traffic entering the network must equal the traffic originating at the router, and the traffic leaving the router must equal the traffic whose destination is the router.

We compare three models. The first one does not consider any uncertain parameters and simplifies the model to only the variables in bold with coefficient 1. The traffic between routers takes a single fixed path, as implemented in [8]. The second model extends the first one with uncertain parameters but without the parameter dependency constraints. The third one is our approach with the parameter dependency constraints added. A parameter constraint, over the flow  $F_{AB}$ , for instance, states that the coefficients representing one given route of traffic from  $A$  to  $B$  take the same value; and the sum of coefficients corresponding to different routes equals to 1. Note that the uncertain parameter equality constraints are already taken into account in the link traffic constraints. The uncertain parameters relative to flow distributions are commonly assumed between 30 and 70 % [12].

Link traffic constraints:

$$\begin{array}{l}
 [\lambda_{1AB}, \lambda_{1AC}, \lambda_{1AD}, \lambda_{2AB}, \lambda_{2AC}, \lambda_{2AD}] \in 0.3..0.7 \\
 A \rightarrow B \quad \lambda_{1AB} * \mathbf{F}_{AB} + \lambda_{1AC} * F_{AC} + \lambda_{1AD} * F_{AD} \quad = 309.0..328.0 \\
 B \rightarrow A \quad \mathbf{F}_{BA} + \mathbf{F}_{CA} + \mathbf{F}_{DA} + \lambda_{1BC} * F_{BC} \quad = 876.39..894.35 \\
 A \rightarrow C \quad \lambda_{2AC} * \mathbf{F}_{AC} + \lambda_{2AD} * \mathbf{F}_{AD} + \lambda_{2AB} * F_{AB} + \lambda_{1BC} * F_{BC} = 591.93..612.34 \\
 B \rightarrow C \quad \lambda_{2BC} * \mathbf{F}_{BC} + \mathbf{F}_{BD} + \lambda_{1AC} * F_{AC} + \lambda_{1AD} * F_{AD} \quad = 543.30..562.61 \\
 C \rightarrow B \quad \lambda_{2AB} * F_{AB} + \mathbf{F}_{CB} + \mathbf{F}_{CA} + \mathbf{F}_{DA} + \mathbf{F}_{DB} \quad = 1143.27..1161.06 \\
 C \rightarrow D \quad \mathbf{F}_{CD} + \mathbf{F}_{BD} + \mathbf{F}_{AD} \quad = 896.11..913.98 \\
 D \rightarrow C \quad \mathbf{F}_{DC} + \mathbf{F}_{DB} + \mathbf{F}_{DA} \quad = 842.09..861.35
 \end{array}$$

Parameter constraints

$$\lambda_{1AB} + \lambda_{2AB} = 1, \lambda_{1AC} + \lambda_{2AC} = 1, \lambda_{1AD} + \lambda_{2AD} = 1, \lambda_{1BC} + \lambda_{2BC} = 1$$

Traffic conservation constraints

$$\begin{array}{l}
 A \text{ origin} \quad F_{AD} + F_{AC} + F_{AB} \quad = 912.72..929.02 \\
 A \text{ destination} \quad F_{DA} + F_{CA} + F_{BA} \quad = 874.70..891.00 \\
 B \text{ origin} \quad F_{BD} + F_{BC} + F_{BA} \quad = 845.56..861.86 \\
 B \text{ destination} \quad F_{DB} + F_{CB} + F_{AB} \quad = 884.49..900.79 \\
 C \text{ origin} \quad F_{CD} + F_{CB} + F_{CA} \quad = 908.28..924.58 \\
 C \text{ destination} \quad F_{DC} + F_{BC} + F_{AC} \quad = 862.53..878.83 \\
 D \text{ origin} \quad F_{DC} + F_{DB} + F_{DA} \quad = 842.0..859.0 \\
 D \text{ destination} \quad F_{CD} + F_{BD} + F_{AD} \quad = 891.0..908.0
 \end{array}$$

We first run the initial model and reproduced the results of [12] in constant time. By adding the uncertain parameters the solution bounds got much larger as the space of potential solutions expanded. However when we run simulations using our approach and the linear EPLEX solver, and we were not able to find any solution to the model with dependency constraints. This shows the importance of taking into account such dependencies, indicating that the data provided match a single path routing algorithm for the sigcomm4 topology. After enlarging the interval bounds of the input data we were able to find a solution with a 50 % split of traffic, but none with 40 – 60 or other combinations. Our approach

showed the effectiveness and strong impact of taking into account dependency constraints with simulations.

## 5 Conclusion

In this paper we introduced an approach to account for dependency constraints among data parameters in an uncertain constraint problem. The approach follows an iterative process that first satisfies the dependency constraints using a branch and bound search. The solutions are then embedded to generate a set of CSPs to be solved. However this does not indicate the relationship between the dependent consistent parameters and possible solutions. We propose to use regression analysis to do so. The current case study showed that by embedding constraint dependencies only one instance had a solution. This was valuable information on its own, but limited the use of regression analysis. Further experimental studies are underway with applications in inventory management, problems clearly permeated with data uncertainty. Even though our approach has been applied to traditional constraint problems in mind, its benefits could be stronger on data mining applications with constraints [6].

## References

1. Benhamou F. and Goualard F. Universally quantified interval constraints. In *Proc. of CP-2000*, LNCS 1894, Singapore, 2000.
2. Boukezzoula R., Galichet S. and Bissierier A. *A MidpointRadius approach to regression with interval data* International Journal of Approximate Reasoning, Volume 52, Issue 9, 2011.
3. Brown K. and Miguel I. *Chapter 21: Uncertainty and Change* Handbook of Constraint Programming Elsevier, 2006.
4. Cheadle A.M., Harvey W., Sadler A.J., Schimpf J., Shen K. and Wallace M.G. *ECLiPSe: An Introduction*. Tech. Rep. IC-Parc-03-1, Imperial College London, London, UK.
5. Chinneck J.W. and Ramadan K. Linear programming with interval coefficients. *J. Operational Research Society*, 51(2):209–220, 2000.
6. De Raedt L., Mannila H., O’Sullivan and Van Hentenryck P. organizers. Constraint Programming meets Machine Learning and Data Mining *Dagstuhl seminar 2011*.
7. Fargier H., Lang J. and Schiex T. Mixed constraint satisfaction: A framework for decision problems under incomplete knowledge. In *Proc. of AAAI-96*, 1996.
8. Medina A., Taft N., Salamatian K., Bhattacharyya S. and Diot C. *Traffic Matrix Estimation: Existing Techniques and New Directions*. Proceedings of ACM SIGCOMM02, 2002.
9. Rossi F., van Beek P., and Walsh T. *Handbook of Constraint Programming*. Elsevier, 2006.
10. Saad A., Gervet C. and Abdennadher S. *Constraint Reasoning with Uncertain Data using CDF-Intervals* Proceedings of CP’AI-OR, Springer, 2010.
11. Van Hentenryck P., Michel L. and Deville Y. *Numerica: a Modeling Language for Global Optimization* The MIT Press, Cambridge Mass, 1997.
12. Yorke-Smith N. and Gervet C. *Certainty Closure: Reliable Constraint Reasoning with Uncertain Data* ACM Transactions on Computational Logic 10(1), 2009.