REPUBLIC OF TUNISIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY OF SFAX
FACULTY OF ECONOMICS AND MANAGEMENT SCIENCES

Towards Statistical Consistency for Stochastic Constraint Programming

PHD THESIS
In Quantitative Methods

by
Imen ZGHIDI

Supervised by
Pr.Dr. Abdelwaheb Rebaï

Academic year
2015–2016

Towards Statistical Consistency for Stochastic Constraint Programming

To my parents, my husband, my children, my siblings, and the extended family members for their everlasting love, support, and encouragements!

Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Dr. Abdelwaheb Rebaï for his guidance, support and encouragement. Besides, I am indebted to Dr. Armagan Tarim, Dr. Roberto Rossi, and Dr. Steven Prestwich for their significant feedback which considerably improved this work. Furthermore, I am grateful to Dr. Yacine Laalaoui for helping me implement and experiment with the algorithms elaborated in this thesis. Last and by no means least, I am very thankful to my husband and my parents for all the love, support and happiness they brought me.

ABSTRACT

In most industrial contexts, decisions are made based on incomplete information. This is due to the fact that decision makers cannot be certain of the future behavior of factors that will affect the outcome resulting from various options under consideration. *Stochastic Constraint Satisfaction Problems* provide a powerful modeling framework for problems in which one is required to take decisions under uncertainty. In these stochastic problems, the uncertainty is modeled by using *discrete random variables* to capture uncontrollable factors like the customer demands, the processing times of machines, house prices, etc. These discrete random variables can take on a set of possible different values, each with an associated probability and are useful to model factors that fall outside the control of the decision maker who only knows the probability distribution function of these random variables which can be forecasted, for instance, by looking at the past behavior of such factors. There are controllable variables on which one can decide, named *decision variables* which allow to model the set of possible choices for the decisions to be made. Finally, such problems comprise *chance constraints* which express the relationship between random and decision variables that should be satisfied within a satisfaction probability threshold – since finding decisions that will always satisfy the constraints in an uncertain environment is almost impossible.

If the random variables' support set is infinite, the number of scenarios would be infinite. Hence, finding a solution in such cases is impossible in general. In this thesis, within the context of an infinite set of scenarios, we propose a novel notion of *statistical consistency*. Statistical consistency lifts the notion of consistency of deterministic constraints to infinite chance constraints. The essence of this novel notion of consistency is to be able to make an inference, in the presence of infinite scenarios in an uncertain environment, based on a restricted finite subset of scenarios with a certain confidence level and a threshold error. The confidence level is the probability that characterises the extent to which our inference, based on a subset of scenarios, is correct whereas the threshold error is the error range that we can tolerate while making such an inference. The statistical consistency acknowledges the fact that making a perfect inference in an uncertain environment and with an infinite number of scenarios is impossible.

The statistical consistency, thus, with its reliance on a limited number of scenarios, a confidence level, and a threshold error constitutes a valid and an appropriate practical road that one can take in order to tackle infinite chance constraints.

We design two novel approaches based on confidence intervals to enforce statistical consistency as well as a novel third approach based on hypothesis testing. We analyze the various methods theoretically as well as experimentally. Our empirical evaluation shows the weaknesses and strengths of each of the three methods in making a correct inference from a restricted subset of scenarios for enforcing statistical consistency. Overall, while the first two methods are able to make a correct inference in most of the cases, the third is a superior, effective, and robust one in all cases.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this Chapter, we introduce the topic of this thesis in Section 1.1. Then, we define our research problem in Section 1.2. Next, in Section 1.3 we summarize the list of our contributions. Finally, we provide an outline of the thesis in Section 1.4.

1.1 Topic of the Thesis

In most industrial contexts, decisions are made based on incomplete information. This is due to the fact that decision makers cannot be certain of the future behavior of factors that will affect the outcome resulting from various options under consideration.

Stochastic Constraint Satisfaction Problems (SCSPs) provide a powerful modeling framework for problems in which we are required to take decisions under uncertainty. In these stochastic problems, the uncertainty is modeled by using *discrete random variables* to capture uncontrollable factors like the customer demands, the processing times of machines, house prices, etc. These discrete random variables can take on a set of possible different values, each with an associated probability and are useful to model factors that fall outside the control of the decision maker who only knows the probability distribution function of these random variables which can be forecasted, for instance, by looking at the past behavior of such factors. We also have controllable variables on which we can decide, named *decision variables* which allow us to model the set of possible choices for the decisions to be made. Finally, such problems comprise *chance constraints* which express the relationship between random and decision variables that should be satisfied within a satisfaction probability threshold – since finding decisions that will always satisfy the constraints in an uncertain environment is almost impossible.

An m-stage SCSP [Wal02, TMW06, HRTP12] is a 7-tuple $\langle V, S, D, P, C, \beta, L\rangle$, where V is a set of decision variables and S is a set of random variables, D is a function mapping each element of V (respectively, S) to a domain (respectively, support) of

potential values. In classical SCSPs both decision variable domains and random variable supports are assumed to be finite. P is a function mapping each element of S to a probability distribution for its associated support. C is a set of chance-constraints over a non-empty subset of decision variables and a subset of random variables. β is a function mapping each chance-constraint $h \in C$ to β_h which is a threshold value in the interval $(0, 1]$. $L = [\langle V_1, S_1\rangle, \ldots, \langle V_i, S_i\rangle, \ldots, \langle V_m, S_m\rangle]$ is a list of *decision stages* such that each $V_i \subseteq V$, each $S_i \subseteq S$, the V_i form a partition of V, and the S_i form a partition of S.

To solve an m-stage SCSP, an assignment to the variables in V_1 must be found such that, given random values for S_1, assignments can be found for V_2 such that, given random values for S_2, ..., assignments can be found for V_m so that, given random values for S_m, the chance constraints are satisfied in the specified fraction of all possible scenarios. Under the assumption that random variable supports are finite, the solution of an m-stage SCSP is, in general, represented by means of a *policy tree* [TMW06]. The arcs in such a policy tree represent values observed for random variables whereas nodes at each level represent the decisions associated with the different stages. Solving an m-stage SCSP is a computationally challenging task and it is in PSPACE [Wal02], in general.

An m-stage SCOP is an m-stage SCSP with an additional objective function f over a non-empty subset of decision and a subset of random variables. An *optimal* solution to a maximization SCOP is a satisfying policy tree with a maximum expected value for f.

An example of a single-stage SCOP is the *stochastic knapsack problem* in which we have a knapsack of capacity k, a set of n items. Each item i has a random weight w_i and a deterministic value v_i. The objective is to find an optimal subset of items so that our chances of exceeding the capacity does not exceed a given threshold β_h. Figure 1.1 shows a SCSP model of the problem with a single chance constraint. For each item i, we introduce a binary decision variable x_i which takes the value 1 if and only if item i is selected to be put in the knapsack, 0 otherwise. We also have a random variable w_i for each item i to model the uncertain weight of the item. The support of each w_i is a set of possible weights, each with a specified probability. The only chance constraint specifies that the probability of not exceeding the knapsack's capacity k should not be below threshold β_h. Finally, the objective function maximises the expected profit.

If the random variables' support set is infinite, the number of scenarios would be infinite and the policy tree itself would be infinite. Hence, finding a satisfying policy tree in such cases is impossible in general. Let us illustrate this situation using the following single-stage SCSP.

Objective function:

$Max\{E\{\sum_{i=1}^{n} v_i x_i\}\}$

Constraints:

$pr\{\sum_{i=1}^{n} w_i x_i \leq k\} \geq \beta_h$

Decision variables:

$x_i \in \{0, 1\} \quad \forall i \in 1, \ldots, n$

Random variables:

$w_i \in support(w_i) \quad \forall i \in 1, \ldots, n$

Stage structure:

$V_1 = \{x_1, x_2, \ldots, x_n\}$

$S_1 = \{w_1, w_2, \ldots, w_n\}$

$L = [\langle V_1, S_1 \rangle]$

Figure 1.1: A single-stage stochastic knapsack model.

Example. Let us consider a single-stage SCSP in which $V = \{x_1, x_2\}$ and $S = \{s_1, s_2\}$. The random variable s_1 may take two possible values 4 and 5, each with probability 0.5. The random variable s_2 may also take two possible values 3 and 4, each with probability 0.5. The domain of the decision variable x_1 is $\{1, 2, 3, 4\}$ whereas the domain of x_2 is $\{3, 4, 5, 6\}$. We have two chance constraints h_1 and h_2 defined as follows:

$$h1 : pr\{s_1 x_1 + s_2 x_2 \geq 30\} \geq 0.75$$

$$h2 : pr\{s_2 x_2 = 12\} \geq 0.5$$

Since this SCSP is a single-stage one, then decision variables x_1 and x_2 must be set to unique values before the random variables are observed. A satisfying policy tree to this SCSP is shown in Figure 1.2 in which $x_1 = 3$ and $x_2 = 6$. The first chance constraint is satisfied in all four scenarios and hence the satisfaction probability is 1 which is larger than the needed 0.75. The second chance constraint is only satisfied in two scenarios out of four with satisfaction probability 0.5 which is enough to make the chance constraint satisfied.

Now, assume instead that the random variables s_1 and s_2 are two *continuous* random variables. Assume s_1 takes values following a uniform distribution over the interval $[4, 5]$ and s_2 takes values following a uniform distribution over the interval $[3, 4]$. Indeed, the policy tree for the same assignment $x_1 = 3$ and $x_2 = 6$ is an *infinite* policy tree as shown in Figure 1.3. It becomes impossible to check whether or not such a policy tree is a satisfying policy tree in a finite amount of time since we need to check an infinite

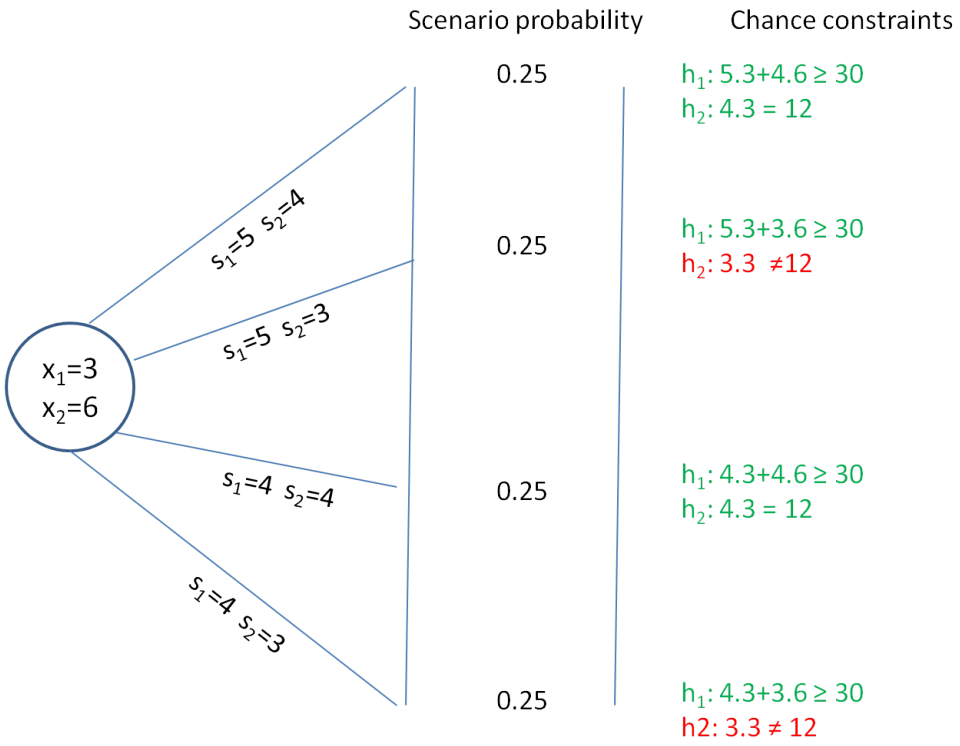

Figure 1.2: A satisfying policy tree with $x_1 = 3$ and $x_2 = 6$.

number of scenarios.

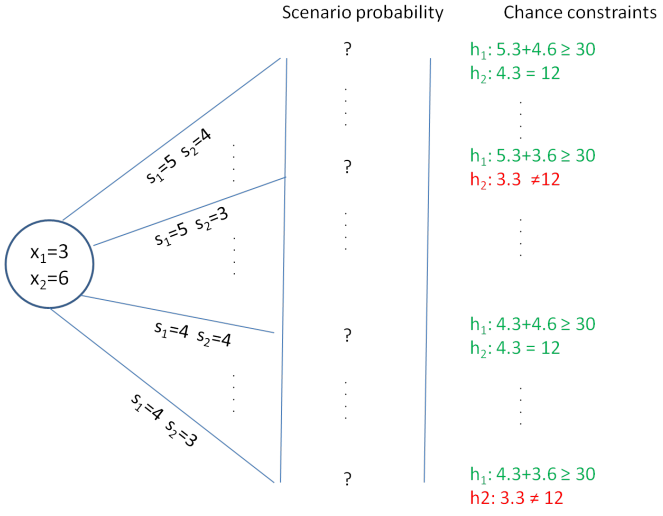

Figure 1.3: An infinite policy tree.

It is important to note that a solution to the classical m-stage SCSP is indeed a trustworthy solution as it considers all possible scenarios and the satisfying policy tree

hedges against all uncertainties. However, it suffers from two main drawbacks: (1) No satisfying policy tree can be found if the random variables are *continuous* since the number of scenarios would have be infinite in that case; and (2) it becomes computationally very challenging as the number of scenarios grows to a certain level so that the resulting models will be too huge to even fit in memory.

The authors in [RHTP11, RHTP15] introduce the notion of (α_c, ϑ)-solutions where α_c[1] is a confidence level and ϑ is a threshold tolerance. Instead of looking for an exact solution, as in the classical m-stage SCSPs, which might not be even possible, we search for a solution that, *with confidence level* α_c, guarantees a satisfaction probability that is no lower than $\beta_h - \vartheta$ for each chance constraint h. The (α_c, ϑ)-solution to an infinite SCSP P is indeed a solution (i.e., a satisfying policy tree) to a restricted version $\hat{P}$ of P in which we only consider a finite subset of the scenarios. In addition to being able to address the drawback of an infinite policy tree, it also provides a more practical framework for decision makers to state their needs through the notions of α_c and ϑ. Furthermore, the method proposed in [RHTP11, RHTP15] provides likelihood guarantees for the quality of the estimated solution, with respect to (wrt) the complete set of scenarios, computed from a finite set of samples.

1.2 Research Problem

The approach in [RHTP11, RHTP15] is a reformulation approach which can be described as follows. Consider an m-stage SCSP P:

$$\boxed{P = \langle V, S, D, P, C, \beta, L \rangle}$$

Note that if any of the random variables in S has an infinite support, then the corresponding policy tree of P would compromise an infinite number of scenarios. No solution method to date can solve such kind of problems in general as it requires finding a satisfying infinite policy tree. Let Ω denote the set of all scenarios of P. The approach in [RHTP11, RHTP15] introduces the novel notion of an (α_c, ϑ)-solution which is a solution to reformulation of P constructed by solving a restricted version of P, $\hat{P}$, by considering a finite subset of the scenarios $\hat{\Omega} \subseteq \Omega$ instead. It is shown in [RHTP11, RHTP15] how to construct the finite set $\hat{\Omega}$ through sampling in such a way that a satisfying policy tree to $\hat{P}$ is indeed an (α_c, ϑ)-solution to P. Thus, the approach in [RHTP11, RHTP15] shows how to find an (α_c, ϑ)-solution to P by reformulating into $\hat{P}$:

[1] In the original paper α_c is referred to as α. But, to differentiate it later on from the significance level α, we rename it here as α_c.

$$\boxed{\hat{P} = \langle V, S, \hat{D}, \hat{P}, C, \beta, L \rangle}$$

such that all the supports are finite and a solution to $\hat{P}$ can be computed using the techniques in [HRTP12] which itself re-use the classical constraint solvers that are implemented for deterministic problems.

However, for certain values of α_c and ϑ, the minimum number of samples required to find an (α_c, ϑ)-solution of an infinite SCSP P may still be too large so that solving the resulting reformulated $\hat{P}$ would be a very challenging task.

In this thesis, we take a different perspective towards computing an (α_c, ϑ)-solution of an infinite single-stage SCSP P. Instead of reformulating the original infinite SCSP into a finite sampled one and use the classical search algorithms based on inference rules suitable for deterministic constraints, we propose to directly work with the infinite one by lifting the inference that guides the search in the deterministic case to *stochastic inference*.

To this end, we need to consider each chance constraint h of an infinite SCSP P:

$$\boxed{h : pr\{C_h\} \geq \beta_h}$$

and propose how to lift the notion of consistency of deterministic constraints to *statistical consistency* for infinite chance constraints. Statistical consistency is a novel concept that we propose in the presence of uncertainty in SCSPs in which the number of scenarios is infinite. The essence of this novel notion of consistency is to be able to make an inference in the presence of infinite scenarios in an uncertain environment based on a restricted finite subset of scenarios (i.e., a scenario sample of restricted size) with a certain confidence level and a threshold error. The confidence level is the probability that characterises the extend to which our inference, based on a subset of scenarios, is correct whereas the threshold error is the error range that we can tolerate while making such an inference. The statistical consistency acknowledges the fact that making a perfect inference in an uncertain environment and with an infinite number of scenarios is impossible. The statistical consistency, thus, with its reliance on a limited number of scenarios, a confidence level, and a threshold error constitutes a valid and an appropriate practical road that one can take in order to tackle infinite chance constraints. Hence, a core research question that we address in this thesis is the following:

How to extend the notion of consistency to an *infinite chance* constraint?

Once the notion of statistical consistency is defined, one needs to also show how to enforce it so that constraint solver can be lifted to *stochastic constraint solvers*. Thus, a second core question that we address in this thesis is the following:

How to enforce statistical consistency?

If one can answer these two core questions in a satisfactory manner, then any constraint solver or search algorithm can use the stochastic inference (through the statistical consistency enforcing algorithm) to directly reason and solve an infinite SCSP in order to compute an (α_c, ϑ)-solution.

1.3 Summary of Contributions

The contributions of this thesis can be summarised as follows:

1. The introduction of the novel notion of statistical consistency, (α_c,ϑ)-consistency, for infinite stochastic constraint satisfaction problems;
2. Two novel approaches based on confidence intervals to enforce (α_c,ϑ)-consistency;
3. A novel third effective and robust approach based on hypothesis testing that is superior to the previous ones based on confidence intervals;
4. An empirical evaluation of the various methods for enforcing (α_c,ϑ)-consistency.

1.4 Outline of the Thesis

- In Chapter 2, we review Stochastic Programming, Stochastic Constraint Programming, and (α_c,ϑ)-solutions in stochastic constraint satisfaction problems.
- In Chapter 3, we propose a novel notion of statistical consistency for stochastic constraint satisfaction problems. This statistical consistency is crucial if one wants to lift the inference about value consistency in the presence of uncertainty, especially when the set of possible scenarios is infinite.
- In Chapter 4, we propose and experimentally validate two novel approaches based on confidence intervals that enforce statistical consistency for stochastic constraint satisfaction problems.
- In Chapter 5, we take a different approach based on composite hypothesis testing that can be used to enforce statistical consistency and validate it experimentally. The results show that it is an effective, robust, and practical method.
- In Appendix A, we present the necessary formal background which includes description of the basic concepts in probability theory and statistics and various sampling methods.

- In Appendix B, we present the implementation details of the key building blocks of our experiments for the approaches based on confidence intervals and hypothesis testing in the $\mathcal{R}$ Language.

Chapter 2

Literature Review

In this chapter, we first introduce Stochastic Programming in Section 2.1. Then, we describe Stochastic Constraint Programming in Section 2.2. Finally, we review (α_c,ϑ)-solutions in stochastic constraint satisfaction problems in Section 2.3.

2.1 Stochastic programming

Stochastic programming is a framework for modeling optimization problems that involve uncertainty [BL97, KW94, Sen72, GW74, AW70, Kuh05]. Examples are scheduling jobs under uncertain processing times, sequencing aircraft landing under random plane departure/arrival delays, inventory control under uncertain future demands (see examples: [Sil78, Por02, Sca60, Ask81, BT88, GS77, Gra99, SWdK04, TS08, TK06, Tem07, PS08]).

To solve optimization problems under uncertainty, stochastic programming needs to represent uncertain elements of these problems. To this end, *random variables* are used as the modeling tool to represent this uncertainty [Vaj72]. Thus, uncertain elements in stochastic programming are assigned to a known probability distribution. For example, a customer demand that is uncertain can be modeled as a random variable with known probability distribution such as a normal distribution.

The typical requirement in stochastic programs is to maintain certain constraints, called *chance constraints*[CC59, CC63]. Chance constraints are constraints that in addition to decision variables also involve random variables and need to be satisfied at a prescribed level of probability called *threshold*. For instance, since customer demand is uncertain, a demand chance constraint can be used to specify that regardless of the demand uncertainty we need to make decision in such a way that the customer demand is guaranteed to be satisfied in 95% of the possible future realizations of the customer demand or simply at threshold 0.95.

	Wheat	Corn	Sugar Beets
Yield (T/acre)	2.5	3	20
Planting cost ($/acre)	150	230	260
Selling price ($/T)	170	150	36 under 6000 T
			10 above 6000 T
Purchase price ($/T)	238	210	
Minimum requirement (T)	200	240	
Total available land: 500 acres			

Table 2.1: Data for farmer's problem

The objective is typically related to minimization (or maximization) of some *expectation* on the problem costs (or profits).

Let us consider a typical optimization problem under uncertainty which is a framer's stochastic problem [BL97]. Throughout this example we will present the basic foundation of stochastic programming and highlight the main advantage of the stochastic programming solution over the classical deterministic approaches.

2.1.1 The Deterministic Farmer's Problem

Consider a farmer who has 500 acres of land. On his land, the farmer can plant three types of crops: grain, corn, and sugar beets. In winter time, the farmer needs to decide how to partition his land among the three different crops such that:

1. At least 200 tons (T) of wheat and 240 T of corn are needed to be used for cattle feed. These amounts can be raised on the farm or bought from a wholesaler. Any production in excess of the feeding requirement would be sold. Selling prices are 170 and 150 per ton of wheat and corn, respectively. The purchase prices are 40% more than this due to the wholesaler's margin and transportation costs;

2. The sugar beet crop sells at 36/T. However, there is a rule that imposes a quota on sugar beet production. Any amount in excess of the quota can be sold only at 10/T. The farmer's quota for next year is 6000 T;

3. Based on past experience, the farmer knows that the mean yield on his land is roughly 2.5 T, 3 T, and 20 T per acre for wheat, corn, and sugar beets, respectively.

In Table 2.1, we summarize the data and the planting costs for these crops.

Minimize:	$150x1 + 230x2 + 260x3 + 238y1 - 170w1$ $+210y2 - 150w2 - 36w3 - 10w4$
Constraints:	$x1 + x2 + x3 \leq 500$ $2.5x1 + y1 - w1 \geq 200$ $3x2 + y2 - w2 \geq 240$ $w3 + w4 \leq 20x3$ $w3 \leq 6000$
Decision Variables:	$x1, x2, x3, y1, y2, w1, w2, w3, w4 \geq 0$

Figure 2.1: A LP model of the deterministic farmer's problem.

The deterministic farmer's problem can be modeled as a Linear Program (LP) [Wol98] as follows.

- $x1$ = acres of land devoted to wheat;
- $x2$ = acres of land devoted to corn;
- $x3$ = acres of land devoted to sugar beets;
- $w1$ = tons of wheat sold;
- $y1$ = tons of wheat purchased;
- $w2$ = tons of corn sold;
- $y2$ = tons of corn purchased;
- $w3$ = tons of sugar beets sold at the favorable price; and
- $w4$ = tons of sugar beets sold at the lower price.

A LP model of the deterministic farmer's problem is shown in Figure 2.1.

The optimal solution to the deterministic farmer's problem based on expected yields is shown in Table 2.2.

This optimal solution is easy to understand. The farmer devotes enough land to sugar beets to reach the quota of 6000 T. He then devotes enough land to wheat and corn production to meet the feeding requirement. The rest of the land is devoted to wheat production. Some wheat can be sold. To an extent, the optimal solution follows a very simple heuristic rule: to allocate land in order of decreasing profit per acre. In this example, the order is sugar beets at a favorable price, wheat, corn, and sugar beets at the lower price. This simple heuristic would, however, no longer be valid if other side constraints, such as labor requirements or crop rotation, would be included.

Culture	Wheat	Corn	Sugar Beets
Surface (acres)	120	80	300
Yield (T)	300	240	6000
Sales (T)	100		6000
Purchase			
Overall profit: 118,600			

Table 2.2: Optimal solution for the deterministic problem

2.1.2 The Shortcomings of the Deterministic Model

Unfortunately, the yields of the crops vary according to the weather. Past experience shows that we have quite different yields for the same crop over different years mainly because of changing weather conditions. Most crops need rain during the few weeks after seeding or planting, then sunshine is welcome for the rest of the growing period. Sunshine should, however, not turn into drought, which causes severe yield reductions. Dry weather is again beneficial during harvest. From all these factors, yields varying 20% to 25% above or below the mean yield are not unusual.

The above deterministic formulation only considers the mean yield which does not take into consideration the possibility of weather conditions. This makes the solution sensitive to any change in the weather conditions. The farmer is left with no sensible decision if the yield varies 20% above or 25% below. Even though the deterministic model can be solved to optimality, this optimal solution is of no use if the weather conditions did not guarantee the mean yield.

So, how can we remedy this problem? How can we incorporate the weather uncertainty when trying to solve the farmer's stochastic problem?

One approach introduced by stochastic programming is to represent these variable yields using discrete and correlated random variables.

A first possibility is to assume some correlation among the yields of the different crops. A very simplified representation of this would be to assume, e.g., that years are good, fair, or bad for all crops, resulting in above average, average, or below average yields for all crops. To fix these ideas, "above" and "below" average indicate a yield 20% above or below the mean yield given in Table 2.1. For simplicity, we assume that weather conditions and yields for the farmer do not have a significant impact on prices. The farmer wishes to know whether the optimal solution is sensitive to variations in yields. He decides to run two more optimizations based on above average and below average yields. Tables 2.3 and 2.4 give the optimal solutions he obtains in these cases. Again, the solutions in Tables 2.3 and 2.4 seem quite natural. When

Culture	Wheat	Corn	Sugar Beets
Surface (acres)	183.33	66.67	250
Yield (T)	550	240	6000
Sales (T)	350		6000
Purchase			
Overall profit: 167,667			

Table 2.3: Optimal solution: average yields (+20%)

Culture	Wheat	Corn	Sugar Beets
Surface (acres)	100	25	375
Yield (T)	200	60	6000
Sales (T)			6000
Purchase		180	
Overall profit: 59,950			

Table 2.4: Optimal solution: average yields (-20%)

yields are high, smaller surfaces are needed to raise the minimum requirements in wheat and corn and the sugar beet quota. The remaining land is devoted to wheat, whose extra production is sold. When yields are low, larger surfaces are needed to raise the minimum requirements and the sugar beet quota. In fact, corn requirements cannot be satisfied with the production, and some corn must be bought.

The optimal solution is very sensitive to changes in yields. The optimal surfaces devoted to wheat range from 100 acres to 183.33 acres. Those devoted to corn range from 25 acres to 80 acres and those devoted to sugar beets from 250 acres to 375 acres. The overall profit ranges from 59,950 to 167,667.

2.1.3 A Scenario Representation

Since long-term weather forecasts cannot be accurately predicted six months ahead, the farmer must make up his mind without perfect information on yields.

The main issue here is clearly on sugar beet production. Planting large surfaces would make it certain to produce and sell the quota, but would also make it likely to sell some sugar beets at the unfavorable price. Planting small surfaces would make it likely to miss the opportunity to sell the full quota at the favorable price. The farmer now realizes that he is unable to make a perfect decision that would be best in all circumstances.

The farmer has to decide now on how to partition the land before observing the realization of the yields in the future. However, the sales and purchases would depend on the outcome of yields. He would, therefore, want to assess the benefits and losses of each decision in each possible future situation or scenario. There are three scenarios in total: above average, average, and below average.

Taking into account the yield uncertainty, one can remodel the Farmer's problem as follows. Decisions on land assignment ($x1$, $x2$, $x3$) have to be taken now, but sales and purchases (wi, $i = 1, \ldots, 4$, yj, $j = 1, 2$) depend on the yields. It is useful to index those decisions by a scenario index $s = 1, 2, 3$ corresponding to above average, average, or below average yields, respectively. This creates a new set of variables of the form wis, $i = 1, 2, 3, 4$, $s = 1, 2, 3$ and yjs, $j = 1, 2$, $s = 1, 2, 3$. As an example, $w32$ represents the amount of sugar beets sold at the favorable price if yields are average. Assuming the farmer wants to maximize long-run profit, it is reasonable for him to seek a solution that maximizes his *expected profit*. If the three scenarios have an equal probability of 1/3, the farmer's problem can be modelled as shown in Figure 2.2

The optimal solution of the model of Figure 2.2 is given in Table 2.5. The top line gives the planting areas, which must be determined before realizing the weather and crop yields. This decision is called the *first stage*. The other lines describe the yields, sales, and purchases in the three scenarios. They are called the *second stage*. The bottom line shows the overall expected profit. The optimal solution can be understood as follows. The most profitable decision for sugar beet land allocation is the one that always avoids sales at the unfavorable price even if this implies that some portion of the quota is unused when yields are average or below average. The area devoted to corn is such that it meets the feeding requirement when yields are average. This implies sales are possible when yields are above average and purchases are needed when yields are below average. Finally, the rest of the land is devoted to wheat. This area is large enough to cover the minimum requirement. Sales then always occur. This solution illustrates that it is impossible, under uncertainty, to find a solution that is ideal under all circumstances. Selling some sugar beets at the unfavorable price or having some unused quota is a decision that would never take place with a perfect forecast. Such decisions can appear in a stochastic model because decisions have to be balanced or *hedged against* the various scenarios.

The hedging effect has an important impact on the expected optimal profit. Suppose yields vary over years but are cyclical. A year with above average yields is always followed by a year with average yields and then a year with below average yields. The farmer would then take optimal solutions as given in Table 2.3, then Table 2.2, then

Minimize:	$150x1 + 230x2 + 260x3$	$-\frac{1}{3}(170w11 - 238y11 + 150w21$ $-210y21 + 36w31 + 10w41)$ $-\frac{1}{3}(170w12 - 238y12 + 150w22$ $-210y22 + 36w32 + 10w42)$ $-\frac{1}{3}(170w13 - 238y13 + 150w23$ $-210y23 + 36w33 + 10w43)$
Constraints:	$x1 + x2 + x3 \leq 500$ $3x1 + y11 - w11 \geq 200$ $3.6x2 + y21 - w21 \geq 240$ $w31 + w41 \leq 24x3$ $w31 \leq 6000$ $2.5x1 + y12 - w12 \geq 200$ $3x2 + y22 - w22 \geq 240$ $w32 + w42 \leq 20x3$ $w32 \leq 6000$ $2x1 + y13 - w13 \geq 200$ $2.4x2 + y23 - w23 \geq 240$ $w33 + w43 \leq 16x3$ $w33 \leq 6000$	
Decision Variables:	$x, y, w \geq 0$	

Figure 2.2: An LP model of the stochastic farmer's problem.

Table 2.4, respectively. This would leave him with a profit of 167,667 the first year, 118,600 the second year, and 59,950 the third year. The mean profit over the three years (and in the long run) would be the mean of the three figures, namely 115,406 per year. Now, assume again that yields vary over years, but on a random basis. If the farmer gets the information on the yields before planting, he will again choose the areas on the basis of the solution in Table 2.2, 2.3, or 2.4, depending on the information received. In the long run, if each yield is realized one third of the years, the farmer will get again an expected profit of 115,406 per year. This is the situation under perfect information. As we know, the farmer unfortunately does not get prior information on the yields. So, the best he can do in the long run is to take the solution as given by Table 2.5. This leaves the farmer with an expected profit of 108,390. The difference between this figure and the value, 115,406, in the case of perfect information, namely 7016, represents what is called the *expected value of perfect information* (EVPI) which represents the loss of profit due to the presence of uncertainty.

Another approach the farmer may have is to assume expected yields and always to

		Wheat	Corn	Sugar Beets
First stage	Area (acres)	170	80	250
$s = 1$ Above	Yield (T)	510	288	6000
	Sales (T)	310	48	6000 (favor. price)
	Purchase (T)	-	-	-
$s = 2$ Average	Yield (T)	425	240	5000
	Sales (T)	225	-	5000 (favor. price)
	Purchase (T)	-	-	-
$s = 3$ Below	Yield (T)	340	192	4000
	Sales (T)	140	-	000 (favor. price)
	Purchase (T)	-	48	-
Overall profit: 108,390				

Table 2.5: Optimal solution based on the stochastic model in Figure 2.2

allocate the optimal planting surface according to these yields, as in Table 2.2. This approach represents the *expected value solution*. It is common in optimization but can have unfavorable consequences. Using the expected value solution every year results in a long run annual profit of 107,240. The loss by not considering the random variations is the difference between this and the stochastic model profit from Table 2.5. This value, 108,390-107,240=1,150, is the *value of the stochastic solution* (VSS), the possible gain from solving the stochastic model. Note that it is not equal to the expected value of perfect information, and may in fact be larger than the EVPI. These two quantities give the motivation for stochastic programming in general. EVPI measures the value of knowing the future with certainty while VSS assesses the value of knowing and using distributions on future outcomes. Our emphasis will be on problems where no further information about the future is available so the VSS becomes more practically relevant. In some situations, however, more information might be available through more extensive forecasting, sampling, or exploration. In these cases, EVPI would be useful for deciding whether to undertake additional efforts.

2.2 Stochastic Constraint Programming

2.2.1 Constraint Satisfaction Problems

Combinatorial Optimization problems are ubiquitous in industry[Tsa93, RvBW06]. Examples are production planning subject to demand and resource availability so that

profit is maximized, scheduling jobs on machines subject to precedence constraints so that the makespan is minimized, vehicle routing subject to initial and final location of the goods and the transportation vehicles so that delivery time and fuel expenses are minimized, etc. Such types of interesting problems arising from real-life can be represented as *constraint satisfaction problems* (CSPs).

A CSP consists of a set of variables, each with a finite domain of values, and a set of constraints specifying allowed combinations of values for some variables [Tsa93]. A *solution* to a CSP is an assignment of variables to values in their respective domains such that all of the constraints are satisfied.

Constraint propagation techniques are inference methods that help reducing the original CSP into another which is smaller in size. The *size* of a CSP is the product of the domain sizes of all variables. To achieve this, each constraint in the original CSP is associated with a constraint propagator. The propagator removes from the domains of the variables in the scope of this constraint *inconsistent* values. Inconsistent values are values that do not appear in any solution to that particular constraint. Inconsistent values can be inferred by using a number of local consistency concepts. For example, a constraint c is *generalized arc consistent* (GAC) iff when a variable is assigned any of the values in its domain, there exist compatible values in the domains of all the other variables of c [MM88] . A CSP is GAC iff all constraints are GAC.

Example. Consider the constraint:

$$4x + 3y - 2z = 10$$

where the domains of the variables are:

$$D(x) = D(y) = D(z) = \{0, 1, \ldots, 9\}$$

Enforcing GAC prunes all the inconsistent values from the domains of x, y, and z, i.e., all the values that do not appear in any solution to the constraint. Thus, enforcing GAC reduces the domains to:

$$D(x) = \{0, 1, \ldots, 7\} \quad D(y) = \{0, 2, 4, 6, 8\} \quad D(z) = \{0, 1, \ldots, 9\}$$

Consistency, however, is neither a necessary nor a sufficient condition for a problem to be solvable. That is why search is needed in order to find a solution to a CSP. Constraint solvers typically, using a search method, explore partial assignments enforcing a local consistency property. Thus, a solution to a CSP is typically found through the interaction of a *search* procedure and *constraint propagation* techniques. In [Tsa93], there are different search algorithms that may be used together with constraint propagation to solve CSPs.

Constraints:

$$(1) \quad \forall j \in V \quad \sum_{i \in B} X_{i,j} = r$$
$$(2) \quad \forall i \in B \quad \sum_{j \in V} X_{i,j} = k$$
$$(3) \quad \forall j_1, j_2, j_1 < j_2 \in V \quad \sum_{i \in B} X_{i,j_1} * X_{i,j_2} = \lambda$$

Decision Variables:

$$X_{i,j} \in \{0, 1\}, \quad \forall i \in V, \forall j \in B$$

Figure 2.3: A constraint program of the BIBD problem

2.2.2 Constraint Programming

Many problems can be modelled as CSPs and efficiently solved by applying the techniques developed for CSPs. *Constraint programming* (CP) provides a platform for users that helps them describe their problems as CSPs.

The CP languages provide constructs for declaring the variables, their domains, and constraints between these variables [HSD98]. In most of these languages the solving techniques are hidden in a solver that is composed of a search algorithm and a propagation algorithm for each constraint.

We now show a typical constraint program for the **balanced incomplete block design** (BIBD) problem which is a standard combinatorial problem from design theory with applications in experimental design and cryptography (prob028 in www.csplib.org). BIBD generation is to find a set of $b > 0$ subsets of a set V of $v \geq 2$ elements such that:

- each subset consists of exactly k elements ($v > k > 0$);
- each element appears in exactly r subsets ($r > 0$);
- each pair of elements appear simultaneously in exactly λ subsets ($\lambda > 0$).

A BIBD instance is thus explained by its parameters $\langle v, b, r, k, \lambda \rangle$.

A BIBD instance is specified by a 2-dimenisonal 0/1 matrix X of $B \times V$, where $B = \{0, \ldots, b-1\}$. A variable $X_{i,j}$ in this matrix takes the value 1 iff the subset i contains the element j. The constraints therefore enforce exactly r 1s per row, k 1s per column, and a scalar product of λ between any pair of distinct rows (see Figure 2.3).

2.2.3 Stochastic Constraint Satisfaction Problems

There have been many attempts to incorporate uncertainty within the CSP formalism (e.g., [THPR08, FLMCS95, BS06, BM06, HRTP11]). But, in this thesis, we focus on the Stochastic Constraint Satisfaction approach.

Stochastic constraint satisfaction problems (SCSPs) are a powerful modeling framework for decision making under uncertainty. SCSPs were first introduced in [Wal02] and further extended in [TMW06] to allow multiple chance-constraints and a range of different objectives. In contrast to classical constraint programming, stochastic constraint programming features random variables and chance-constraints, i.e. constraints that should be satisfied according to a prescribed probability.

An m-stage SCP is defined as a 7-tuple $\langle V, S, D, P, C, \beta, L\rangle$, where

- $V = \{x_1, x_2, \ldots, x_m\}$ is a set of decision variables;
- $S = \{s_1, s_2, \ldots, s_m\}$ is a set of stochastic variables;
- D is a function mapping each element of V and each element of S to a domain of potential values. Both decision and stochastic variable domains are assumed to be finite;
- P is a function mapping each element of S to a probability distribution for its associated domain;
- C is a set of *chance-constraints* over a non-empty subset of decision variables and a subset of stochastic variables;
- β is a function mapping each chance-constraint $h \in C$ to β_h which is a threshold value in the interval $(0, 1]$. We will denote a chance-constraint by using the notation "$\Pr\{\langle cons\rangle\} \geq \beta_h$", meaning that constraint $\langle cons\rangle$, constraining decision and random variables, should be satisfied with probability greater or equal to β_h; and
- $L = [\langle V_1, S_1\rangle, \ldots, \langle V_i, S_i\rangle, \ldots, \langle V_m, S_m\rangle]$ is a list of *decision stages* such that each $V_i \subseteq V$, each $S_i \subseteq S$, the V_i form a partition of V, and the S_i form a partition of S. The decision stages are defined based on a *response policy*. A response policy states the rules that decide when decision variables have to be set.

The solution of an m-stage SCP is, in general, represented by means of a *policy tree* [HRTP12]. The arcs in such a policy tree represent values observed for stochastic variables whereas nodes at each level represent the decisions associated with the different stages. Each level of the tree represents a stage.

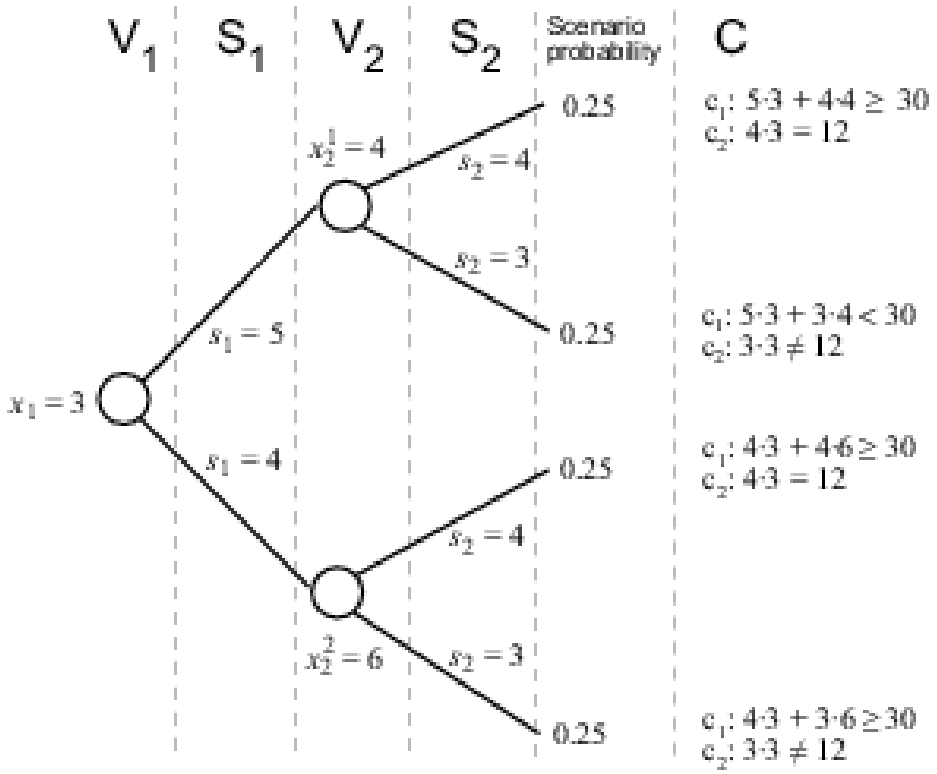

Figure 2.4: Policy tree for the SCSP in Example 1

In a policy tree, an assignment to the variables in V_1 must be found such that, given random values for stochastic variables in S_1, an assignment can be found for V_2 such that, given random values for S_2, ... , an assignment can be found for V_m so that , given random values for S_m the chance-constraints be satisfied with probability greater or equal to β_h.

Unlike SP, SCP offers a richer modeling language which supports chance-constraints over global, nonlinear, logical constraints in addition to linear ones, as well as an elegant way of describing the stage structure. Detailed comparisons between the Artificial Intelligence approach and the Operations Research approach to modeling and solving integer combinatorial problems[NW88] in general can be found in [HC88, LP01].

Example. Let us consider a two-stage SCSP ([HRTP12]) in which $V_1 = \{x_1\}$ and $S_1 = \{s_1\}$, $V_2 = \{x_2\}$ and $S_2 = \{s_2\}$. Stochastic variable s_1 may take two possible values, 5 and 4, each with probability 0.5; stochastic variable s_2 may also take two possible values, 3 and 4, each with probability 0.5. The domain of x_1 is $\{1, \ldots, 4\}$ and the domain of x_2 is $\{3, \ldots, 6\}$. There are two chance constraints in C, $c_1 : \Pr\{s_1x_1 + s_2x_2 \geq 30\} \geq 0.75$ and $c_2 : \Pr\{s_2x_1 = 12\} \geq 0.5$. In this case, the decision variable x_1 must be set to a unique value before random variables are observed, while decision variable x_2 takes a value that depends on the observed value of the random

variable s_1. A possible solution to this SCSP is the satisfying policy tree shown in Fig. 2.4 in which $x_1 = 3, x_2^1 = 4$ and $x_2^2 = 6$, where x_2^1 is the value assigned to decision variable x_2, if random variable s_1 takes value 5, and x_2^2 is the value assigned to decision variable x_2, if random variable s_1 takes value 4.

As the example shows, a solution to a SCSP is not simply an assignment of the decision variables in V to values, but it is instead a satisfying policy tree. For a recent survey on different approaches to uncertainty see [HRTP10].

2.3 (α_c,θ)-Solutions

In [RHTP11], the authors argue that in the presence of infinite set of scenarios due to continuous random variables, the complete existing approaches in [TMW06, HRTP12] cannot be employed directly to solve such stochastic problems. There are a number of ways to overcome this problem. For instance, different sampling strategies have been proposed in [TMW06] in order to reduce a-priori the support of random variables and therefore produce SCSPs that are manageable in size. Taking a different heuristic approach, the authors in [PTRH09] build a neural network to encode a policy function that takes the best possible decision with respect to the past history of decisions taken and values observed for the random variables. Nevertheless, their approach is purely heuristic and does not provide any likelihood guarantee on the quality of the solutions found. In contrast, [RHTP11] shows that statistical estimation via confidence intervals can be employed to provide likelihood guarantees for the quality of the solutions found when using N samples rather than all the scenarios. In fact, in the presence of continuous random variables, the decision maker can only hope to "estimate" a solution in general. Therefore, the concept of (α_c,ϑ)-solutions introduced two more parameters that can be set by the decision maker that provide certain likelihood guarantees. Given, a SCSP, an (α_c,ϑ)-solution is an assignment $\mathcal{A}$ in which for each chance constraint $\overline{h}$ (where $\overline{h} : pr\{\overline{c}\} \geq \beta_h$ denotes the chance constraint h in which the decision variables take the values in assignment $\mathcal{A}$), with confidence level α_c, we guarantees that the satisfaction probability of $\overline{c}$ (i.e, $pr\{\overline{c}\}$) to be no less that $\beta_h - \vartheta$.

Definition 1. *(in [RHTP11, RHTP15]) An (α_c,θ)-solution to a SCSP is an assignment that at least with probability α_c provides for every chance constraint $h \in C$ a satisfaction probability greater than or equal to $\beta_h - \vartheta$.*

In operations research, and particularly in stochastic programming, the state-of-the-art technique that applies sampling in combinatorial optimisation is the sample average approximation (SAA) method [KSHDM01, ASS02, LSW06, Bra12a, Bra12b].

SAA methods for problems comprising a single chance constraint were discussed in [AS08, LA08, PAS09, Che52, Hoe63, Bra12a, Bra13].

Chapter 3

Statistical Consistency

In this chapter, we propose a novel notion of statistical consistency for stochastic constraint satisfaction problems. This statistical consistency is crucial if one wants to lift the inference about value consistency in the presence of uncertainty, especially when the set of possible scenarios is infinite. The new notion of statistical consistency is parameterised by a confidence probability, α_c, and a threshold error ϑ. The goal is to be able to make an inference, with confidence probability α_c, about whether a value is consistent or not with respect to the uncertain environment without the error exceeding threshold value ϑ. This new statistical consistency can be used to develop new family of search algorithm that enforce such consistency during search. By doing so, we achieve to lift the reasoning and inference about value consistency during search for stochastic constraint satisfaction problem in which we have infinite set of scenarios.

The rest of this Chapter is organised as follows. In Section 3.1, we introduce some basic search algorithms for the CSP and show how to combine search and propagation in Section 3.2 in order to boost the search process. Then, we formalise the notion of (α_c,ϑ)-consistency in Section 3.3 and propose for the first time a novel concept of statistical consistency to stochastic constraint satisfaction problems. Finally, we conclude in Section 3.4.

3.1 CSP: Search Algorithms

Recall that a CSP consists of a set of variables, each with a finite domain of values, and a set of constraints specifying allowed combinations of values for some variables [Tsa93]. A *solution* to a CSP is an assignment of variables to values in their respective domains such that all of the constraints are satisfied.

As an example, consider the Map Coloring Problem [Tsa93] in which we are given a map such as the one in Figure 3.1 and three colors red, green, and blue. The objective

is to color every region with a color such that no two neighboring regions are colored with the same color.

Figure 3.1: A Map Coloring Problem.

A solution to the Map Coloring Problem in Figure 3.1 is shown in Figure 3.2.

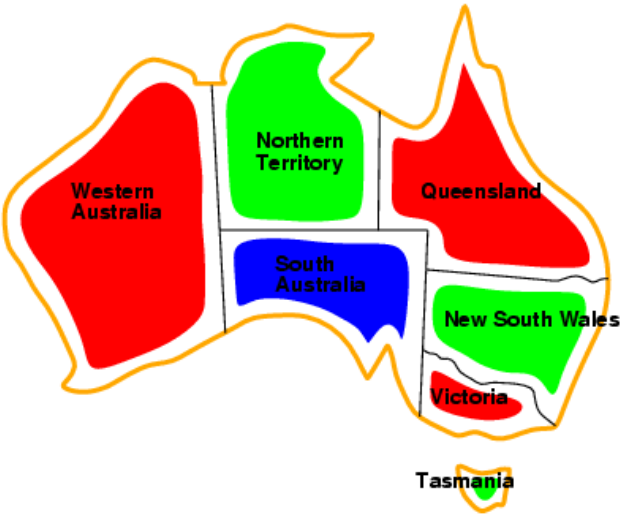

Figure 3.2: A Solution to a Map Coloring Problem.

Modeling the Map Coloring problem as a CSP consists of decalring a set of decision variables, their domains and a set of constraints. In Figure 3.3, we show the encoding of the decision variables: each region is mapped to a decision variable. In Figure 3.4. we show the constraint network which states a binary not-equal constraint between every two decision variables whose corresponding regions are neighbors. The domain of each

decision variable is made up of the three colors. It is obvious to see that if there exists an assignment of values to variables such that all constraints are satisfied represents a solution to the Map Coloring problem.

Figure 3.3: Decision Variables of the Map Coloring Problem.

How to *search* for a solution to a CSP? To search is to make an (educated) guess among several alternatives. But be prepared to undo that guess and try a different alternative if the guess does not lead to a solution. We will focus on systematic search. Given sufficient time, if there is a solution, it will be found. If there is no solution, the search space will be exhausted and the search will report that there is no solution.

The typical search process when solving a CSP is searching through a space of partial assignments. A *partial assignment* is an assignment to one or more decision variables. We generally begin with an empty assignment and incrementally attempt to extend it into a solution. We will assume a backtracking search style: If we discover that the current partial assignment cannot be extended to a solution (a dead end), we backtrack over last decision made and try an alternative one.

The search for a solution to a CSP may be viewed as exploring a *tree*(see Figure 3.5). The root of the tree represents the CSP before any search choices. The choices made correspond to *branches* in the tree. The descendants of the root node correspond to sub-CSPs, i,e., the original CSP augmented with a partial assignment which leaves us with a simplified problem.

There are two common branching styles used. In *d-way branching*, each branch under a parent node represents the assignment of one of the d domain values from the domain of a particular variable as depicted in Figure 3.6.

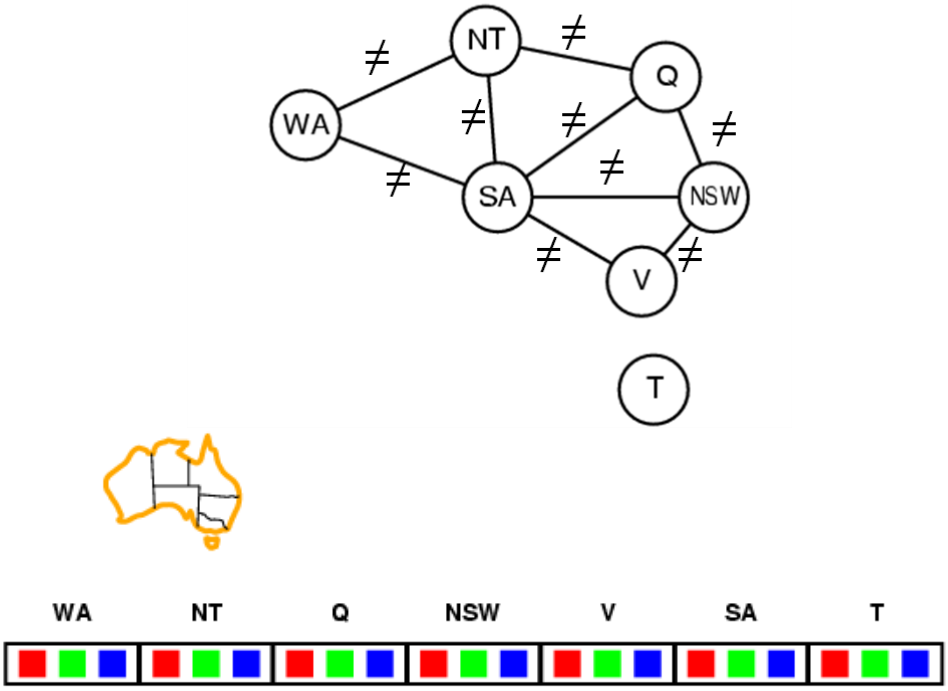

Figure 3.4: A Constraint Model of the Map Coloring Problem.

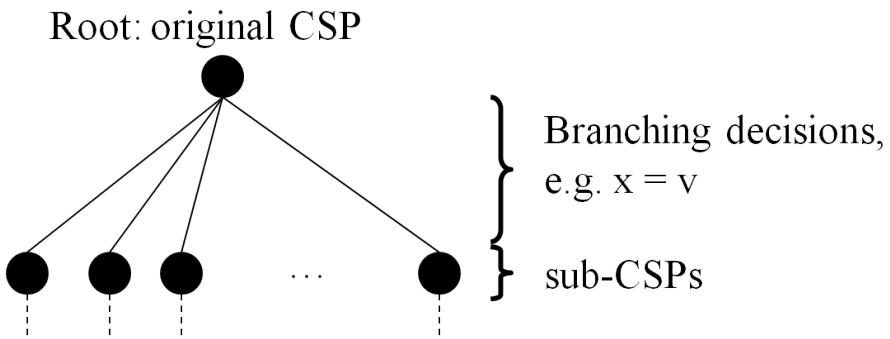

Figure 3.5: Search as Tree Traversal.

In, *2-way branching*, we try extending partial assignment with $x = v$ first. If no solution found, we remove v from consideration before continuing as depicted in Figure 3.7.

Note that most modern constraint solvers [LtOpt94, BPS99, ILO07] use 2-way branching. The *instantiation order* is the order in which assignments to variables are made. The *level* in a search tree corresponds to the number of assignments made and

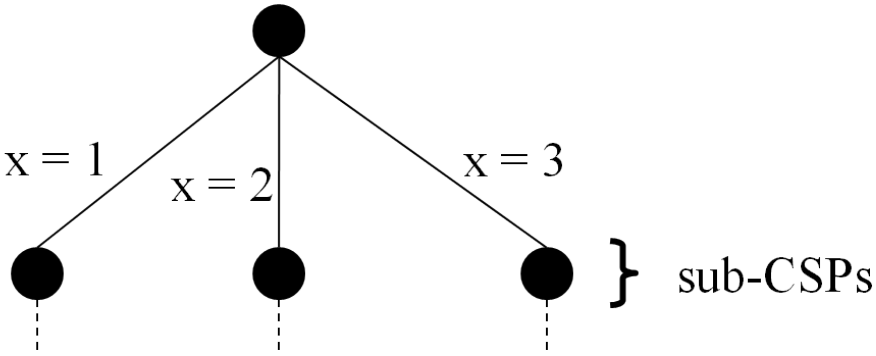

Figure 3.6: D-way Branching.

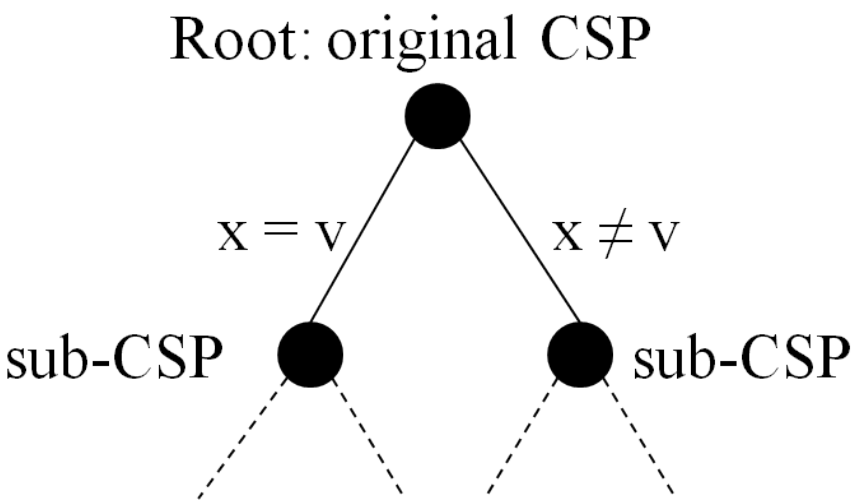

Figure 3.7: 2-way Branching.

also known as *search depth*. During search, we call those variables that have not yet been assigned as *future variables*. Similarly, those variables that have been assigned are called *past variables*. Each branch in a search tree represents a partial assignment (see Fgure 3.8 for an example). A branch that reaches depth n, is a *complete assignment*.

Generate&Test. A simple (but very expensive) method of solving a CSP. Each possible complete assignment is generated and then tested to see if it satisfies all the constraints. It is a brute force method that only checks constraints after a complete assignment has been generated. It is a systematic search algorithm and hence guaranteed to find a solution if one exists (eventually), but it is almost never used in practice.

Backtracking. Backtracking is a general algorithm for finding all (or some) solutions to combinatorial problems, notably constraint satisfaction problems, that incrementally builds candidates to the solutions, and abandons each partial candidate c

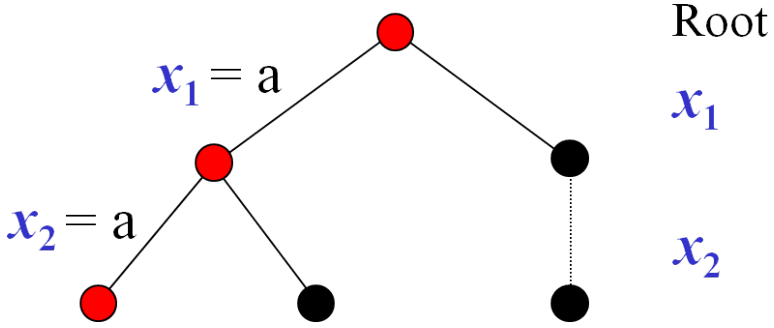

Figure 3.8: Partial assignment $\langle x_1 = a, x_2 = a \rangle$.

("backtracks") as soon as it determines that c cannot possibly be completed to a valid solution [Knu68]. It improves on Generate&Test by incrementally extending partial solutions. Every time we make an assignment, we check to see if a constraint has been violated. The backtracking algorithm traverses the search tree recursively, from the root down, in *depth-first order*. At each node c, the algorithm checks whether c can be completed to a valid solution. If it cannot because a constraint has been violated, the whole sub-tree rooted at c is skipped (*pruned*). Otherwise, the algorithm (1) checks whether c itself is a valid solution if it is a depth n, and if so reports it; and (2) recursively enumerates all sub-trees of c. Therefore, the actual search tree that is traversed by the algorithm is only a part of the potential tree. The total cost of the algorithm is the number of nodes of the actual tree times the cost of obtaining and processing each node.

Part of the Search Tree of the Map Coloring Problem that could be explored using the Backtracking algorithm is shown in Figure 3.9.

The Backtracking algorithm is also systematic, like Generate&Test, and so is guaranteed to find a solution if one exists. Since it checks a constraint as soon as all of the variables that it constrains are instantiated, it can spot dead-ends much faster than Generate&Test. In general, the sooner you can spot a dead-end, the more search you will save.

3.2 Combining Search and Propagation

How can we do better than the Backtracking algorithm? Indeed, the Backtracking algorithm just looks backwards and checks a constraint when all the variables in the scope of such a constraint are instantiated. Hence, the constraints are only used for checking a partial assignment but not as a means to make inference about the values

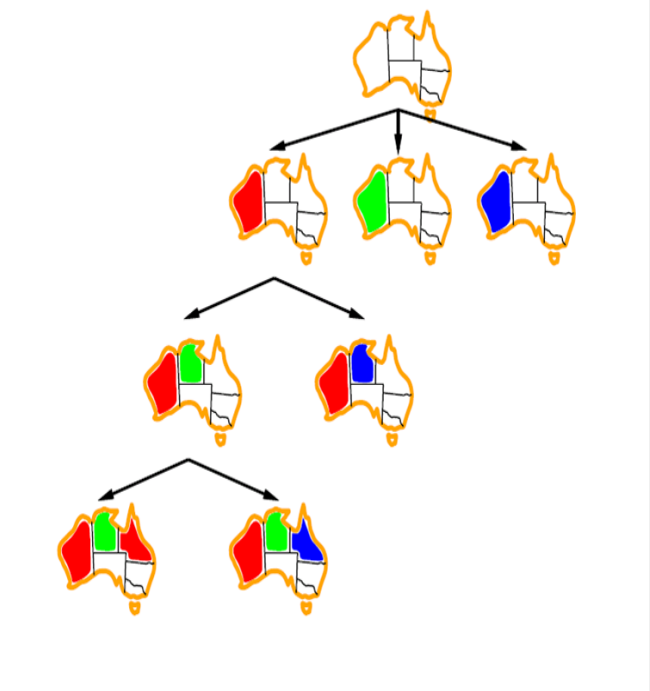

Figure 3.9: Part of the Search Tree of the Map Coloring Problem.

of future variables. As the size of the search space is the product of the domain sizes of all variables and is thus too big in general to enumerate all possible assignments, many CP solution methods are based on *local consistency* techniques which help reduce the size of the search space and hence improving on the Backtracking algorithm. The main idea is to detect and remove from the domains of the future variables all those values that cannot be a part of any solution. Such values lack *support* and are called *inconsistent*.

Constraint propagation is the process of making deduction (or: inference) via a subset of the constraints. The deduced information is recorded as changes to the domains of the future variables by pruning values from domains. Any local change to the domain of any decision variable during search can forms the basis for further deductions. Hence the result of a change is gradually propagated through the constraint network.

A *consistency property* holds when constraint propagation of a certain kind reaches a *fixpoint*, i.e., we can deduce nothing new. Consistency may be (1) *local* involving one constraint or (2) *global* which involves all the constraints. Many local consistency notions exist in the literature [Apt03]. For instance, the following type of consistency are defined for binary constraints:

Definition 2. *A binary constraint is* (i, j)-consistent *iff its variables have non-empty domains and any consistent assignment of* i *variables can be consistently extended to* j

additional variables.

One of the most widely used consistency techniques for binary constraints is arc-consistency (AC) and is defined as follows:

Definition 3. *A binary constraint is* arc-consistent *iff it is* $(1, 1)$*-consistent.*

A CSP typically has several constraints, and a value in a domain may not be a part of a solution even though it is consistent with respect to a local consistency property defined on a single constraint. Hence, consistency is *not a sufficient condition* for a problem to be solvable.

A general notion of arc-consistency, which is not restricted to binary constraints, is generalised arc-consistency (GAC):

Definition 4. *A constraint is* generalised arc-consistent *iff its variables have non-empty domains, and for any value in the domain of a variable, there exist consistent values in the domains of the other variables.*

GAC is the strongest form of consistency and weaker forms of consistency exist such as bounds consistency which is defined only for totally ordered domains. The reader can see [DB97] and [Wal01] for a comparison of different consistency properties.

Algorithms maintaining consistency on the constraints by removing the inconsistent values from the domains [Apt03] are incorporated into CP systems. The fundamental way in which systematic constraint solvers work is as follows. Before searching for solutions, the constraints need to be examined for maintaining consistency. During search, as an assignment to x_i is made, we propagate its consequences. When the domain of a variable x_i is modified, values in the domain of the other variables participating in the same constraint as x_i might lose their support and become inconsistent. If this is the case then the constraint has to be examined and consistency needs to be established if necessary. Propagation may result in removal of inconsistent values, which is also known as *pruning* or *filtering*. This change may lead to further inconsistencies, hence the result of any modification is gradually propagated through the entire CSP. Finally, this process terminates. We then have three possible situations:

1. a domain becomes empty (domain wipe-out) and thus a failure occurs. We backtrack and try an alternative assignment and remove the value that led to this failure from the domain of the decision variable;

2. a solution is found and the search process is terminated;

3. there exists at least one variable which is not ground (i.e. whose domain is not singleton yet) and all the constraints are consistent. Then, we pick a future variable and assign a value for it and the whole process starts over.

See [vB06] for a survey for Backtracking algorithms combined with propagation for CSPs.

An example of how search and propagation are combined towards solving the Map Coloring problem is given in Figure 3.10, Figure 3.11, Figure 3.12, Figure 3.13, and Figure 3.14 in which we discover a dead-end.

In Figure 3.10, when we assign WA to red, the color red is removed, due to constraint propagation, from the domains of the two neighboring variables NT and SA.

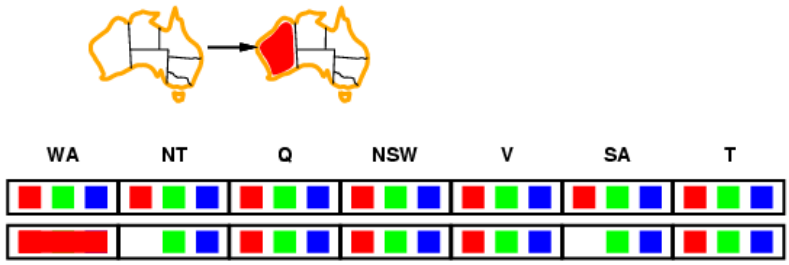

Figure 3.10: Constraint propagation after assigning WA to red.

In Figure 3.11, the effect of propagating assigning Q to green leads to the pruning of the color green from the domains of its neighbor variables NT, NSW, SA. But, the propagation does not stop there. As depicted in Figure 3.12, since the domain of SA is a singleton of color blue, blue is pruned from the domain of NSW and is reduced to a singleton color red. Since NSW is a neighbor of V, then red is pruned from the domain of V as shown in Figure 3.13. Finally, in Figure 3.14, because of the constraint between NT and SA and blue being the only remaining value in the domain of both variables a domain wipe-out occurs due to propagation.

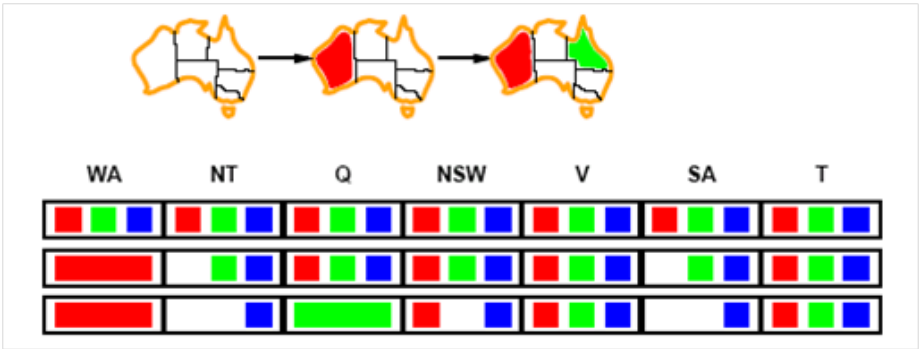

Figure 3.11: Constraint propagation after assigning Q to green (continued).

Bessière and Régin have defined GAC-schema [BR97] which is a general framework for AC algorithms. GAC-schema is based on the AC-7 arc-consistency algorithm for

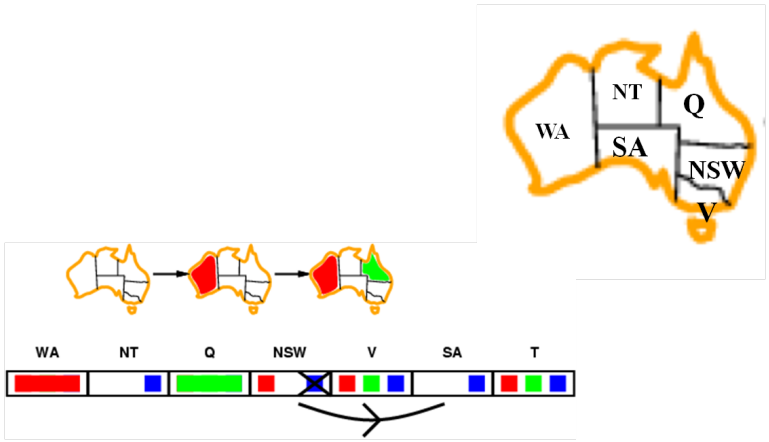

Figure 3.12: Constraint propagation after assigning Q to green (continued).

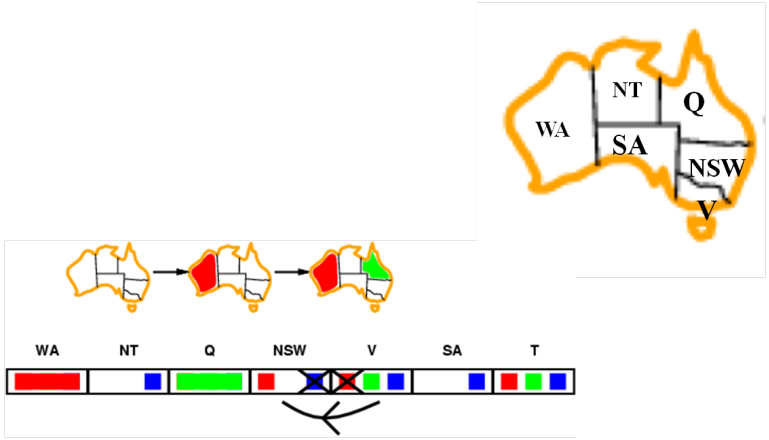

Figure 3.13: Constraint propagation after assigning Q to green (continued).

binary constraints [BFR99] and allows enforcing GAC on constraints of arbitrary arity. Bessière and Régin have also defined a schema for enforcing GAC on an arbitrary conjunction of constraints [BR98].

Many constraints arising in the real-life CSPs are often non-binary. A *global constraint* is a constraint which involves usually more than 2 variables, and encapsulates its own *filtering algorithm* which is used to propagate the constraint [BH03]. Many useful global constraints have been proposed in the last ten years [BCDP07]. An example is the *all-different* constraint:

$$\textit{all-different}(\langle X_1, \ldots, X_n \rangle)$$

which holds iff no pair of variables in $\langle X_1, \ldots, X_n \rangle$ are assigned the same value. A specialised efficient and effective pruning algorithm for the *all-different* constraint is

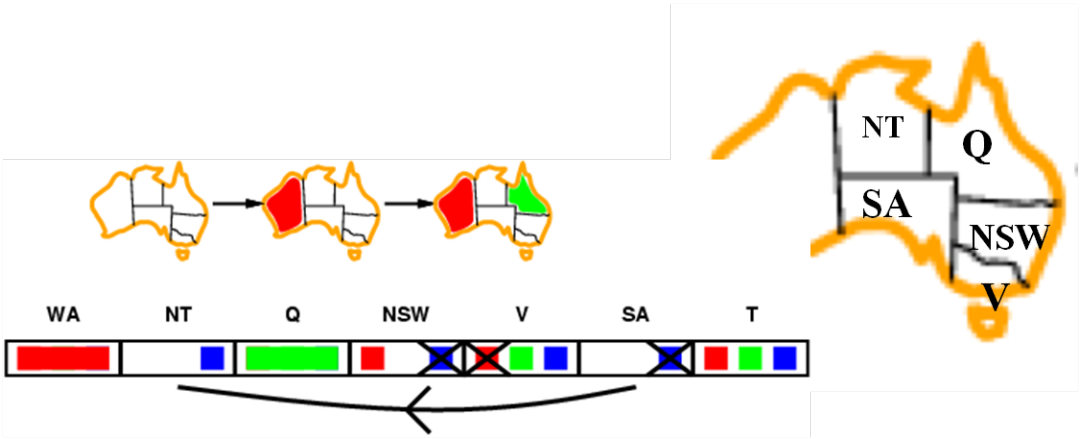

Figure 3.14: Domain Wipe-out: Constraint propagation after assigning Q to green (continued).

given in [Reg94]. It is generally inefficient to decompose such constraints into simple binary constraints because the total pruning obtained by the propagation of each simpler constraint is likely to be weaker. As another example, [Rég96] makes use of the flow theory and proposes an efficient GAC filtering algorithm for the global cardinality constraint.

3.3 (α_c,ϑ)-consistency

Constraint propagation techniques are inference methods that help reducing the original CSP into another which is smaller in size. How can we extend such a notion to *chance constraints* in *SCSPs* in which the random variables have either a *finite support set* or even an *infinite* one?

The authors in [HRTP12] extend the notion of GAC for only global chance constraints in SCSPs in which the random variables have a *finite* support set. Since, in this thesis we restrict ourselves to single-stage SCSPs, we present a simplified definition of GAC for single-stage SCSPs. Let us consider a single stage SCSP $\langle V, S, D, P, C, \beta, L\rangle$. Let $h \in C$ be a chance constraint constraining a subset of variables $X_h \subseteq V$ and a non-empty subset of random variables $S_h \subseteq S$:

$$h : pr\{C_h\} \geq \beta_h$$

Let Ω_h denote the set of scenarios constructed from the random variables S_h. Let A be an assignment of the decision variables in X_h. Let $\mathcal{T}_h^A$ be the policy tree restricted to h, i.e., the policy tree for a single-stage SCSP composed only of the chance constraint h

which is $\langle V_h, S_h, D, P, \{h\}, \beta, \langle V_h, S_h \rangle\rangle$ in which the decision variables X_h take the values in assignment A. In other words, each arc in this policy tree is a possible scenario $s \in \Omega_h$ whose probability is denoted by $pr(s)$. Let C_h^s denote the constraint C_h in which the random variables are replaced with the actual values in s. Let $\overline{C_h^s}$ denote the expression in which the decision variables in C_h^s take the values specified in assignment A. Let the boolean B_h^s be 1 if expression $\overline{C_h^s}$ is satisfied, 0 otherwise.

In [BHHW07], the authors studied the computational complexity of reasoning with global constraints and identified a number of crucial questions. The question that is is at the core of all generic arc consistency algorithms is the question which is generally asked for all values one by one is the following [BHHW07]:

Instance. A constraint C, a domain D on $var(C)$ (i.e., the variables in the scope of constraint C), and a value v for variable $x \in var(C)$

Question. Does value v for x have a support on C in D? I.e., does there exist an assignment A that satisfies C in which $x = v$

How do we establish support for a given value for a *chance constraint*? We propose the following definition:

Definition 5. *Given a chance constraint h. A value v in the domain of $x \in X_h$ is GAC iff there exists an assignment A in which $x = v$ and*

$$\sum_{s \in \Omega_h} B_h^s pr(s) \geq \beta_h$$

In other words, a value v in the domain of $x \in X_h$ is GAC iff there exits an assignment A in which $x = v$ and $\mathcal{T}_h^A$ is a satisfying policy tree.

Definition 6. *A chance constraint h is GAC iff every value in the domain of every variable in X_h is GAC.*

Definition 7. *A SCSP is GAC iff every chance constraint is GAC.*

The authors in [HRTP12] propose novel approximate generic propagation algorithms for any chance constraint that reuse the corresponding propagators of the deterministic version of the chance constraints.

Note that the above consistency definition for chance constraints assumes that all the random variables in S_h have *discrete finite* support. If at least one random variable in S_h has an *infinite* support, then Ω_h would be an infinite set of scenarios and, for any assignment A, the corresponding policy tree $\mathcal{T}_h^A$ would compromise an infinite number of arcs. Indeed, the sum that tests whether a value v is GAC or not would be an infinite sum. So, how to proceed is one of the core questions that this thesis tries to address.

Inspired by the concept of (α_c,ϑ)-solutions proposed in [RHTP15], we introduce the notion of *(α_c,ϑ)-consistency* for chance constraints of single-stage SCSPs in which at least one random variable has an *infinite support*. Due to the infinite number of scenarios that we are required to consider in order to establish whether or not a value v is GAC which makes it practically impossible, instead we seek to establish whether value v is consistent, with confidence α_c, and error tolerance threshold ϑ by looking at a finite subset of sampled scenarios. By having the parameters α_c, we control how confident we want to be in our judgment whereas ϑ controls the level of error we are ready to tolerate. Thus, we are now ready to introduce the novel definition of (α_c,ϑ)-consistency as follows.

Definition 8. *Given a chance constraint h. A value v in the domain of $x \in X_h$ is (α_c, ϑ)-GAC iff there exists an assignment A in which $x = v$ and, with confidence level α_c,*

$$\sum_{s \in \Omega_h} B_h^s pr(s) \geq \beta_h - \vartheta$$

Note that in $100\alpha_c\%$ of the times, a value v who is truly consistent is detected as so according to this definition with error ϑ. In the other cases, we accept the fact that we incorrectly classify a truly consistent value as inconsistent as well as a truly inconsistent value as consistent.

Now consider two values v and w in the domain of $x \in X_h$. If v is (α_c, ϑ)-GAC and w is (α_c, ϑ)-GAC, then together v and w are not (α_c, ϑ)-GAC since the errors accumulate. Indeed, simultaneously v and w are guaranteed to be (α_c^2, ϑ)-GAC. Therefore, unlike the definition of GAC for chance constraints in which random variables are discrete and finite, if a value v is GAC and another value w is GAC, then both are GAC. In our case, we need to worry about multiple statements since our definition of (α_c, ϑ)-consistency is probabilistic in nature. Thus, for a chance constraint we propose the following novel definition:

Definition 9. *A chance constraint h is (α_c, ϑ)-GAC iff* simultaneously *every value in the domain of every variable in X_h is (α_c, ϑ)-GAC.*

Similarly, when we consider a SCSP composed of multiple chance constraints, we propose the following definition:

Definition 10. *A SCSP is (α_c, ϑ)-GAC iff* simultaneously *every chance constraint is (α_c, ϑ)-GAC*

3.4 Conclusion

In this chapter, inspired by the concept of (α_c,ϑ)-solutions proposed in [RHTP15], we introduce , for the first time, the notion of (α_c,ϑ)-consistency for infinite chance constraints for single-stage SCSPs in which at least one random variable has an infinite support. The essence of this novel notion of consistency is to be able to make an inference about value consistency in the presence of infinite scenarios in an uncertain environment based on a restricted finite subset of scenarios (i.e., a sample of restricted size) with a certain confidence level α_c and a threshold error ϑ. The confidence level α_c characterises the extend to which our inference, based on a subset of scenarios, is correct whereas the threshold error ϑ is the error range that we can tolerate while making such an inference. The statistical consistency acknowledges the fact that making a perfect inference in an uncertain environment and with an infinite number of scenarios is impossible. The (α_c,ϑ)-consistency, a form of statistical consistency, with its reliance on a limited number of scenarios, a confidence level, and a threshold error constitutes a valid and an appropriate practical road that one can take in order to tackle inference about value consistency in the context of infinite chance constraints.

Chapter 4

Enforcing Statistical Consistency via Confidence Intervals

In this chapter, in Section 4.1 we review the concept of confidence intervals. Then, in Section 4.2, we present and experimentally validate our first approach to infer whether or not a value v is (α_c,ϑ)-consistent which is an adaptation of the approach in [RHTP15]. By taking a different approach, we show in Section 4.3 our second method which is an improvement over the first one and validate it empirically. Finally, before we conclude in Section 4.5, we show how to use the Bonferroni correction to enforce (α_c,ϑ)-consistency in Section 4.4.

4.1 Confidence Intervals

To draw conclusions about a scientific system, statisticians make use of fundamental laws of probability and statistical inference. The theory of statistical inferences consists of those methods by which one makes inferences or generalizations about a population by using information gathered in the form of samples drawn from the population to estimate some population parameter.

A *p*oint estimator of some population parameter Θ is a single value $\hat{\theta}$ of a statistics $\hat{\Theta}$. For example, the value $\bar{x}$ of the statistics $\bar{X}$ which is the sample mean, computed from a sample of size n is a point estimate of the population mean μ.

It is true that our accuracy increases with large samples, but there is still no reason why we should expect a point estimate from a given population to be a representative of the population.There are many situations in which it is preferable to determine an interval within which we would expect to find the value of the parameter. Such an interval is called an *interval estimate*. An interval estimate is defined by two numbers, between which a population parameter, is said to lie. An interval estimate of a popula-

tion parameter θ is an interval of the form $\hat{\theta_L} < \theta < \hat{\theta_U}$, where $\hat{\theta_L}$ and $\hat{\theta_U}$ depend on the the value of the statistic $\hat{\theta}$ for a particular sample and also on the sampling distribution of $\hat{\theta}$. For example, $a < \mu < b$ is an interval estimate for the population mean μ. It indicates that population mean is greater than a but less than b.

The most prevalent form of interval estimation is *c*onfidence intervals. From the sampling distribution of $\hat{\theta}$, we should be able to determine $\hat{\theta_L}$ and $\hat{\theta_U}$ such that the probability of $\hat{\theta_L} < \theta < \hat{\theta_U}$ is equal to any positive value between 0 and 1. If, for instance, we find $\hat{\theta_L}$ and $\hat{\theta_U}$ such that the probability of $\hat{\theta_L} < \theta < \hat{\theta_U}$ is $1 - \alpha$, for $0 < \alpha < 1$, then we have a probability of $1 - \alpha$ of selecting a random sample that will produce an interval covering θ. The interval $[\hat{\theta_L}, \hat{\theta_U}]$ computed from the selected sample, is then called a $(1 - \alpha)100\%$ *c*onfidence interval, the fraction $1 - \alpha$ is called the *c*onfidence coefficient or the *d*egree of confidence, and the endpoints, $\hat{\theta_L}$ and $\hat{\theta_U}$, are called the *l*ower and *u*pper confidence limits, respectively.

When α is 0.05, we have a 95% confidence interval. But, when α is 0.01, we obtain a *w*ider 99% confidence interval. Indeed, the more confident the confidence interval is the wider it is. But of course, it is better to be 95% confident that the average life of a certain machine is between 7 and 8 years than to be 99% confident that it is between 3 and 10 years old, Ideally, we prefer a short interval with a high degree of confidence level.

Since, in Chapter 4 of this thesis we will focus on binomial distribution, we will illustrate next one of the exact methods to construct confidence intervals. The Clopper-Pearson interval [CP34] is a common method for calculating binomial confidence intervals. The Clopper-Pearson interval is an exact interval since it is based directly on the binomial distribution rather than any approximation to the binomial distribution. This interval never has less than the nominal coverage for any population proportion, but that means that it is usually conservative. For example, the true coverage rate of a 95% Clopper-Pearson interval may be well above 95%, depending on n and the width of the interval.

The Clopper-Pearson interval is a symmetric two-sided confidence interval but can be expressed as a single-sided interval:

$(p_{\text{lb}}, 1)$ and $(0, p_{\text{ub}})$ where

$$p_{\text{lb}} = \min\{p | Pr\{\text{bin}(N; p) \geq X\} \geq 1 - \tfrac{\alpha}{2}\},$$
$$p_{\text{ub}} = \max\{p | Pr\{\text{bin}(N; p) \leq X\} \geq 1 - \tfrac{\alpha}{2}\},$$

X is the number of successes (or "yes" events) observed in the sample, $\text{bin}(N; p)$ is a binomial random variable with N trials and probability of success p and α is the significance level. Note that we assume $p_{\text{lb}} = 0$ when $X = 0$ and that $p_{\text{ub}} = 1$ when

$X = N$.

4.2 Statistical Inference Rules: A First Approach

4.2.1 Presenting the First Approach

Suppose we are given a chance constraint $h : pr\{C\} \geq \beta_h$ over an infinite set of scenarios Ω_h, an assignment A in which $x = v$, a confidence level α_c, and a threshold error ϑ. Our goal is to answer the question of whether v is an (α_c, ϑ)-consistent.

Recall that a value v in the domain of $x \in X_h$ is (α_c, ϑ)-consistent iff there exists an assignment A in which $x = v$ and, with confidence level α_c,

$$\sum_{s \in \Omega_h} B_h^s pr(s) \geq \beta_h - \vartheta$$

Let $\mu = \sum_{s \in \Omega_h} B_h^s pr(s)$ be the actual satisfaction probability.

A value v in the domain of $x \in X_h$ is (α_c, ϑ)-inconsistent iff there exists no assignment A in which $x = v$ and, with confidence level α_c,

$$\mu \geq \beta_h - \vartheta$$

For the rest of this section, we suppose we are given a chance constraint $h : pr\{C\} \geq \beta_h$ over an infinite set of scenarios Ω_h, a specific assignment A in which $x = v$, a confidence level α_c, and a threshold error ϑ. Our goal is to answer the question of whether v is an (α_c, ϑ)-consistent with respect to just an assignment A. Later on, we show how v is (α_c, ϑ)-consistent or not with respect to the whole chance constraint h.

In [RHTP15, RHTP11], the authors introduced the concept of an (α_c, ϑ)-solution of a SCSP P in which the set of scenarios Ω is infinite, and where α_c is a confidence level and ϑ is a threshold tolerance error. An (α_c, ϑ)-solution is an assignment that, *with confidence level* α_c, guarantees a satisfaction probability that is no lower than $\beta_h - \vartheta$ for each chance constraint h in P.

Therefore, one can relate the notion of (α_c, ϑ)-consistency and (α_c, ϑ)-solution as follows:

Definition 11. *A value v in the domain of $x \in X_h$ is an (α_c, ϑ)-consistent iff there exists an assignment A in which $x = v$ that is an (α_c, ϑ)-solution to the SCSP consisting of a single chance constraint h.*

The approach in [RHTP15, RHTP11] shows that an (α_c, ϑ)-solution can be found by searching for a solution to a reformulation of P, $\hat{P}$, by considering a *finite* subset of the scenarios $\hat{\Omega} \subseteq \Omega$ instead. The new SCSP $\hat{P}$ is referred to as a *sampled SCSP*. It

is shown in [RHTP11, RHTP15] how to construct the finite set $\hat{\Omega}$ through *sampling* in such a way that a satisfying policy tree to $\hat{P}$ is indeed an (α_c, ϑ)-solution to P. Thus, the approach in [RHTP11, RHTP15] shows how to find an (α_c, ϑ)-solution to P by reformulating into $\hat{P}$ by determining the minimum sample size n that guarantees that.

The approach in [RHTP15, RHTP11] uses the exact Clopper-Pearson confidence interval [CP34] to analytically compute the minimum sample size n required for each chance constraint h to satisfy a given α_c, ϑ, and β_h. The paper also shows that if you find a solution to a SCSP restricted to a sample of size n by using the current existing approaches[TMW06, HRTP12], then that solution is indeed an (α_c,ϑ)-solution to the original problem.

Definition 12. *As stated in [RHTP15], n is computed as the minimum value for which*

$$\max(p_{ub}^{\beta_h} - \beta_h, \beta_h - p_{lb}^{\beta_h}) \leq \vartheta,$$

where $p_{lb}^{\beta_h}$ and $p_{ub}^{\beta_h}$ are the single-sided Clopper-Pearson confidence interval bounds for a confidence probability α_c, and round($\beta_h N$) "successes" in n trials; round() approximates the value to the nearest integer.

According to [RHTP15], if we let $h_1, \ldots, h_k$ be k chance constraints in a SCSP P, let $\hat{P}$ be a sampled SCSP over n samples, where n is the number of samples required to guarantee a confidence level α_c and an error tolerance threshold ϑ for each constraint h_i considered independently, according to the previous definition, then by Proposition 4 in [RHTP15], any policy tree $\hat{\mathcal{T}}$ that is a solution to $\hat{P}$ is an (α_c, ϑ)-solution to P. Note, however according to the contrapositive of proposition 1 in [RHTP15], that if $\mu \in]p_{\text{lb}}^{\beta_h}, p_{\text{ub}}^{\beta_h}[$, then we can neither prove or disprove that a policy tree $\mathcal{T}$ is an (α_c, ϑ)-solution or not. The same is also true when $\mu \in]\beta_h - \vartheta, \beta_h + \vartheta[$ according to contrapositive of proposition 2 in [RHTP15]. If $\mu \geq \beta_h + \vartheta$, then according to proposition 2 in [RHTP15] we have a satisfying policy tree. If $\mu \leq \beta_h - \vartheta$, then according to proposition 2 in [RHTP15] we cannot have a satisfying policy tree.

We are now ready to propose the first method for testing whether a value v is (α_c, ϑ)-consistent or not by using the approach in [RHTP15] as follows:

Step 1: Compute the sample size n required to guarantee a confidence level α_c and an error tolerance threshold ϑ;

Step 2: Solve the sampled SCSP $\hat{\mathcal{P}}$ composed of the single chance constraint h in which the domain of x is set to $\{v\}$ using a sample of scenarios of size n;

Step 3: Classify v as (α_c,ϑ)-consistent if $\hat{\mathcal{P}}$ has a satisfying policy tree. Otherwise, v is classified as (α_c,ϑ)-inconsistent.

$\vartheta = 0.01$	6900
$\vartheta = 0.009$	8500
$\vartheta = 0.008$	10700
$\vartheta = 0.007$	15000
$\vartheta = 0.005$	28000
$\vartheta = 0.001$	700000

Figure 4.1: The theoretical sample size n needed as ϑ varies to guarantee a 0.95 confidence level as per Definition 2 in [RHTP15].

Note that, theoretically, this method should be able to make correct classification in $\alpha_c\%$ of the times as long as $\mu \notin]\beta_h - \vartheta, \beta_h + \vartheta[$.

4.2.2 Validating the First Approach

Now, we run some empirical study to validate the first approach.

Let us consider a chance constraint $Pr\{C\} \geq \beta_h$ over an infinite set of scenarios Ω_h in which all scenarios have the same probability a, and an assignment A in which $x = v$. Let $\mu = pr\{C_A\}$ where C_A is the constraint C in which all decision variables take the values in A. For any given scenario, $s \in \Omega_h$, expression C_A^s denotes C_A in which all the random variables take the values in s. Expression C_A^s is 1 iff it is satisfied, 0 otherwise.

Now,

$$\mu = \sum_{s \in \Omega_h} C_A^s.a$$

Let $\Omega_h = \Omega_h^s \cup \Omega_h^u$ where for each $s \in \Omega_h^s$ we have that C_A^s is 1 and for each $s \in \Omega_h^u$ we have C_A^s is 0.

So when all scenarios have the same probability, we have

$$\mu = \frac{\sum_{s \in \Omega_h^s} C_A^s}{|\Omega_h|} = \frac{|\Omega_h^s|}{|\Omega_h|}$$

Thus, the satisfaction probability (μ) we are trying to estimate, from a set of samples, can be seen as estimating the "success" probability of the associated Bernoulli trial. This variable can produce only two outcomes: "yes" with probability p and "no" with probability $1 - p$. In our case, the value $p = \mu$ — the "yes" probability — is unknown, but can be estimated by repeatedly observing the behavior of the random variable in a sequence of Bernoulli trials. Such an experiment is known as the *binomial experiment* [Pap84].

If we consider a sample S_n of size n, the sample mean $\bar{x}$ is equal to:

$$\bar{x} = \frac{\sum_{s \in S_n^s} C_A^s}{n} = \frac{|S_n^s|}{n}$$

where $S_n = S_n^s \cup S_n^u$ where for each $s \in S_n^s$ we have that C_A^s is 1 and for each $s \in S_n^u$ we have C_A^s is 0. Note that $\bar{x}$ is the binomial proportion of success in the first n trials. The proportion of success $\bar{x}$ is typically used to estimate the probability of success p when this probability is unknown. indeed, the expected value of $\bar{x}$ is p.

To test our first method, we set our true mean μ to 0.5 of a Bernoulli trial. We vary β_h to take values in

$$\{0.48, 0.49, 0.5, 051, 0.52, 0.53\}$$

and

$$\vartheta \in \{0.01, 0.009, 0.008, 0, 007, 0.005, 0.001\}$$

.

In Figure 4.1, we show the theoretical sample size n needed for different values of ϑ to guarantee a 0.95 confidence level as per Definition 2 in [RHTP15]. As expected, as ϑ gets smaller, the sample size increases.

For each sample size n we generate a sample S_n of size n of 0's and 1's where each 0 and 1 is generated using a Bernoulli distribution whose $p = \mu$. Indeed, each value in S_n simulates, with probability μ, the outcome of a scenario in which the chance constraint $h : pr\{C\} \geq \beta_h$ is satisfied, and with probability $1 - \mu$, the outcome of a scenario in which the chance constraint is unsatisfied. The sample mean $\bar{x}$ is the satisfaction probability restricted to the n scenarios. So, if $\bar{x} \geq \beta_h$, we have a satisfying policy tree and hence classify value v as (α_c,ϑ)-consistent. Otherwise, we classify value v as (α_c,ϑ)-inconsistent.

For each configuration $\langle \beta_h, \alpha_c, \vartheta \rangle$ we run 1000 experiments and record the number of times c we classify value v as (α_c,ϑ)-consistent and the number of times f it is (α_c,ϑ)-inconsistent. Now, if the true mean μ is greater than or equal to $\beta_h - \vartheta$, we compute the Correct Classification Rate (CCR) of our method as

$$\frac{c}{c+f} = \frac{c}{1000}$$

Note that, $\frac{f}{f+c}$ represents the ratio of wrong classification in this case. This is the ratio of type I error since we are rejecting a consistent value v. If, however, the true mean μ is strictly smaller than $\beta_h - \vartheta$, then CCR is computed as

$$\frac{f}{c+f} = \frac{f}{1000}$$

	Infer v as (α_c, ϑ)-consistent	Infer v as (α_c, ϑ)-inconsistent
v is truly (α_c, ϑ)-consistent	Correct classification	Wrong classification
v is truly (α_c, ϑ)-inconsistent	Wrong classification	Correct classification

Figure 4.2: Four possible outcomes of an experiment ($\langle S_n, \beta_h, \alpha_c, \vartheta \rangle$)

	Consistent
$\beta_h = 0.48$	**99.8%**
$\beta_h = 0.49$	**95.4%**
$\beta_h = 0.50$	49.1%
$\beta_h = 0.51$	5.5%
	Inconsistent
$\beta_h = 0.52$	**100%**
$\beta_h = 0.53$	**100%**

Figure 4.3: The correct classification rate (CCR) for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.01$ using the first approach.

Note that, $\frac{c}{f+c}$ represents the ratio of wrong classification in this case. This is the ratio of type II error since we are accepting an inconsistent value v.

There are four possible outcomes in each experiment as depicted in Figure 4.2. Indeed, CCR is the most important indicator of whether or not the confidence interval approach is effective or not in correctly classifying (α_c, ϑ)-consistent values or not.

Figures 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 show the results of our experiments.

From the results in Figure 4.3, we are able to make the following remarks:

- The CCR of a consistent value is above 95%, as expected by the theory, when $\mu \geq \beta_h + \vartheta$ (i.e, for $\beta_h = 048$ and $\beta_h = 0.49$ in this case).

- When $\beta_h = 0.50$, we are in the critical case since $\beta_h - \vartheta < \mu < \beta_h + \vartheta$. As expected from our theory, the first inference method cannot guarantee a CCR above or equal to 95%. The first method in this case is only able to achieve a 49.1% CCR.

- When $\mu = \beta_h - \vartheta$, the true classification of the value should be consistent, yet, our inference method classifies it as inconsistent. The CCR in this case is as low as 5.5%. This case requires more attention because according to [RHTP15] (proposition 1) such situations are included in the case of $\mu \leq \beta_h - \vartheta$ which is *wrongly* treated as violating the chance constraint and therefore from the first approach's point of view this value must be classified as inconsistent whereas

	Consistent
$\beta_h = 0.48$	**100%**
$\beta_h = 0.49$	**97.2%**
$\beta_h = 0.50$	49.9%
	Inconsistent
$\beta_h = 0.51$	**96.5%**
$\beta_h = 0.52$	**100%**
$\beta_h = 0.53$	**100%**

Figure 4.4: The correct classification rate (CCR) for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.009$ using the first approach.

	Consistent
$\beta_h = 0.48$	**100%**
$\beta_h = 0.49$	**98%**
$\beta_h = 0.50$	51.2%
	Inconsistent
$\beta_h = 0.51$	**97.2%**
$\beta_h = 0.52$	**100%**
$\beta_h = 0.53$	**100%**

Figure 4.5: The correct classification rate (CCR) for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.008$ using the first approach.

in reality it is not. This kind of flaw will be treated and corrected in our next methods.

- When $\mu \leq \beta_h - \vartheta$, the CCR of an inconsistent value is above 95% as expected by the theory (i.e., for $\beta_h = 0.52$ and $\beta_h = 0.53$ in this case).

The results in Figure 4.4, Figure 4.5, Figure 4.6, Figure 4.7, and Figure 4.8 do confirm the theory:

- The CCR of a consistent value is above 95%, as expected by the theory, when $\mu \geq \beta_h + \vartheta$
- When $\beta_h = 0.50$, we are in the critical case . As expected from our theory, the first inference method cannot guarantee a CCR above or equal to 95%. The first method in this case is only able to achieve around 50% CCR.

	Consistent
$\beta_h = 0.48$	**100%**
$\beta_h = 0.49$	**99.2%**
$\beta_h = 0.50$	51.5%
	Inconsistent
$\beta_h = 0.51$	**98.4%**
$\beta_h = 0.52$	**100%**
$\beta_h = 0.53$	**100%**

Figure 4.6: The correct classification rate (CCR) for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.007$ using the first approach.

	Consistent
$\beta_h = 0.48$	**100%**
$\beta_h = 0.49$	**100%**
$\beta_h = 0.50$	51.7%
	Inconsistent
$\beta_h = 0.51$	**99.9%**
$\beta_h = 0.52$	**100%**
$\beta_h = 0.53$	**100%**

Figure 4.7: The correct classification rate (CCR) for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.005$ using the first approach.

- When $\mu \leq \beta_h - \vartheta$, the CCR of an inconsistent value is above 95% as expected by the theory.

The results confirm the expected theoretical result . Indeed, in every configuration, the CCR is above 95% as expected as long as $\mu \notin]\beta_h - \vartheta, \beta_h + \vartheta[$ and in fact are slightly better in practice. When we are in the critical case, increasing the sample size beyond the theoretical value does not have any positive effect except for the case when $\beta_h = .50$ in Figure 4.3. But, when $\mu = \beta - \vartheta$, the inference method is flawed and unable to make a correct classification and need to be corrected.

Thus, in summary, our first approach is able to achieves a CCR of 95% or above as long as $\mu \notin]\beta_h - \vartheta, \beta_h + \vartheta[$ which leaves us with an important challenge: *Is it possible to do more correct inference in the critical case and overcome the flawed case?* I.e., when $\mu \in [\beta_h - \vartheta, \beta_h + \vartheta[$.

	Consistent
$\beta_h = 0.48$	**100%**
$\beta_h = 0.49$	**100%**
$\beta_h = 0.50$	48.7%
	Inconsistent
$\beta_h = 0.51$	**100%**
$\beta_h = 0.52$	**100%**
$\beta_h = 0.53$	**100%**

Figure 4.8: The correct classification rate (CCR) for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.001$ using the first approach.

4.3 Statistical Inference Rules Through Confidence Intervals: A Second Approach

In this section, we first revisit the previous approach and look at it with a different perspective and gain new insights so that we are able to present our second method. Finally, we empirically validate it.

4.3.1 Looking at the First Approach Using a Different Perspective

Another possible angle from wish we can analyze the first approach is to reason about the value of $\bar{x}$ as an estimated quantity consisting of an interval rather than an exact value. Indeed, in the first approach we do compute $\bar{x}$ using a sample of size n but we do not reason about it as an estimate quantity of μ that may have some error. Instead, the error threshold value ϑ is used to constitute an interval around β_h (i.e, $[\beta_h - \vartheta, \beta_h + \vartheta]$) according to the approach in [RHTP15]. But, β_h is indeed known and constant! What is uncertain is indeed the quantity μ and $\bar{x}$ is the sample mean that constitutes an estimate of this unknown true mean μ. A better way to represent our margin of error is to consider the estimated quantity as an interval $[\bar{x} - \vartheta, \bar{x} + \vartheta]$ rather than an exact value. This interval is called a confidence interval (CI) and can be constructed in many ways to compromise a margin of error with a confidence level α_c for the true mean [Vol93, AC98, CP34]. In our case, we will use, as in [RHTP15], the exact Clopper-Pearson confidence intervals [CP34]. In fact, the bounds $\bar{x} - \vartheta$ and $\bar{x} + \vartheta$ are the single-sided Clopper-Pearson confidence interval bounds for a confidence probability α_c [CP34].

More formally, a chance constraint h, for a given assignment A in which $x = v$, a confidence interval $[\bar{x} - \vartheta, \bar{x} + \vartheta]$ can be computed from a sample of size n such that the coverage probability of this interval of the true mean is at least α_c, i.e., the actual probability that the interval contains the true mean is α_c. The true mean μ, covered by the CI $[\bar{x} - \vartheta, \bar{x} + \vartheta]$ with coverage probability α_c can be any value in that interval.

Let us analyze the cases of where β_h can be with respect to the CI $[\bar{x} - \vartheta, \bar{x} + \vartheta]$, having the first approach's assumption that

$$\mu \notin]\beta_h - \vartheta, \beta_h + \vartheta[$$

in mind:

Clear accept: If $\beta_h \leq \bar{x} - 2\vartheta$, then even when μ takes the minimum possible value $\bar{x} - \vartheta$, we have that

$$\beta_h \leq \bar{x} - 2\vartheta (= \bar{x} - \vartheta - \vartheta = \mu - \vartheta)$$

So, we have $\mu \geq \beta_h + \vartheta$ and, hence, according to proposition 2 of [RHTP15], with probability α_c, assignment A can be proved to be a satisfying policy tree. Thus, v can be classified correctly as being (α_c,ϑ)-consistent. See Figure 4.9 for an illustration.

Clear reject: If $\beta_h \geq \bar{x} + 2\vartheta$, then even when μ takes the maximum possible value $\bar{x} + \vartheta$, we have that

$$\beta_h \geq \bar{x} + 2\vartheta (= \bar{x} + \vartheta + \vartheta = \mu + \vartheta)$$

So, we have $\mu \leq \beta_h - \vartheta$ and, hence, according to proposition 2 of [RHTP15], with probability α_c, assignment A can be proved to be a non-satisfying policy tree. Thus, v can be classified correctly as being (α_c,ϑ)-inconsistent. See Figure 4.10 for an illustration.

Undecidable: If $\beta_h \in]\bar{x} - 2\vartheta, \bar{x} + 2\vartheta[$, then we consider two sub-cases:

Case a: If $\beta_h \in]\bar{x} - 2\vartheta, \bar{x}]$, then $\beta_h + \vartheta$ belongs to the CI $[\bar{x} - \vartheta, \bar{x} + \vartheta]$ and so does the true mean μ. It is not possible to verify whether $\mu \geq \beta_h + \vartheta$ is true or not. Indeed, we are in the case described by the contrapositive of proposition 2 in [RHTP15] and the assumption $\mu \notin]\beta_h - \vartheta, \beta_h + \vartheta[$ is no longer valid. Thus, it is impossible to classify v as being (α_c,ϑ)-(in)consistent. See Figure 4.11 for an illustration.

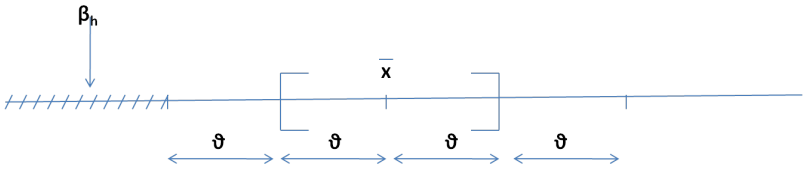

Figure 4.9: The "clear accept" case.

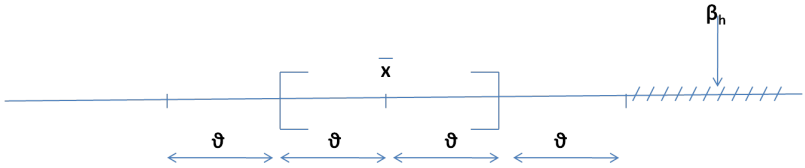

Figure 4.10: The "clear reject" case.

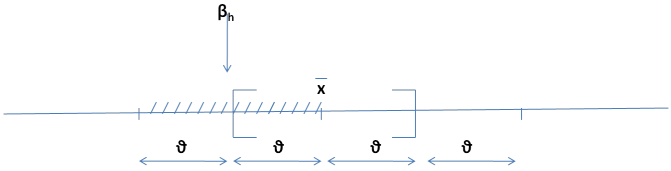

Figure 4.11: The "undecidable" case: sub-case a.

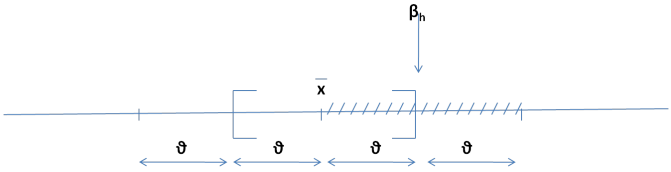

Figure 4.12: The "undecidable" case: sub-case b.

Case b: If $\beta_h \in [\bar{x}, \bar{x}+2\vartheta[$, then $\beta_h - \vartheta$ belongs to the CI $[\bar{x}-\vartheta, \bar{x}+\vartheta]$ and so does the true mean μ. It is not possible to verify whether $\mu \leq \beta_h - \vartheta$ is true or not. Indeed, we are in the case described by contrapositive of proposition 2 in [RHTP15] and the assumption $\mu \notin]\beta_h - \vartheta, \beta_h + \vartheta[$ is no longer valid. Thus, it is impossible to classify v as being (α_c,ϑ)-(in)consistent. See Figure 4.12 for an illustration.

Thus, one can easily design a new inference method as follows:

Step 1: Find a large enough sample size n, compute $\bar{x}$, and construct a confidence interval $[\bar{x} - \vartheta, \bar{x} + \vartheta]$ so that its coverage probability is α_c;

Step 2: We have three mutually exclusive cases:

Consistent: If $\beta_h \leq \bar{x} - 2\vartheta$, then classify v as (α_c,ϑ)-consistent;

Inconsistent: If $\beta_h \geq \bar{x} + 2\vartheta$, then classify v as (α_c,ϑ)-inconsistent;

Mixed: We have two sub-cases:

- If $\beta_h \in]\bar{x} - 2\vartheta, \bar{x}]$, classify v as (α_c,ϑ)-consistent because $\bar{x} \geq \beta_h$;
- If $\beta_h \in]\bar{x}, \bar{x} + 2\vartheta]$, classify v as (α_c,ϑ)-inconsistent because $\bar{x} < \beta_h$;

It remains, however, a challenging task to figure out how to compute a large enough sample size n to guarantee a CI of width 2ϑ. Despite that, the two points of view lead to the same conclusions but using quite two different approaches. Next, however, we will show how to extend the approach based on CIs in order to improve our correct inference. We will also show how to compute the sample size that is large enough to guarantee the proper width of our CI.

4.3.2 Presenting the Second Approach

Let us consider a CI $[\bar{x} - \vartheta, \bar{x} + \vartheta]$ constructed using a sample of size n. Let $min = \bar{x} - \vartheta$ and $max = \bar{x} + \vartheta$. Now, recall that our main goal is to answer the following question: is $\mu \geq \beta_h - \vartheta$ or not?

We make the following number of observations:

Observation 1. If $\beta_h \leq min$, then it is also true that $\beta_h - \vartheta \leq min$. Thus, since μ is covered by the CI $[\bar{x} - \vartheta, \bar{x} + \vartheta]$ with probability α_c and even in the worst case when $\mu = min$ we still have $\mu \geq \beta_h - \vartheta$, with probability α_c, we can classify v as being (α_c,ϑ)-consistent in this case.

Note that, in the first approach, the inference mentioned in Observation 1 is only possible if $\beta_h \leq \bar{x} - 2\vartheta$ as discussed above. Thus, by changing our point of view, we already can make more correct inference, with probability α_c when the first method fails to do so.

Our second observation is about the width of the CI which is 2ϑ. Is it necessary for the width to be 2ϑ?

Observation 2. Let us assume we restrict the width of the CI to be ϑ instead of 2ϑ, i.e., the CI is $[\bar{x} - \frac{\vartheta}{2}, \bar{x} + \frac{\vartheta}{2}]$. Now, consider the case when $\beta_h \in [\bar{x} - \frac{\vartheta}{2}, \bar{x} + \frac{\vartheta}{2}]$. Recall that our true mean is covered by CI with coverage probability α_c. Now, we have two cases: (1) If $\mu \geq \beta_h$, then it is also true that $\mu \geq \beta_h - \vartheta$; (2) if $\mu < \beta_h$, then in the worst case μ can be as small as $min = \bar{x} - \frac{\vartheta}{2}$ and β_h as large as $max = \bar{x} + \frac{\vartheta}{2}$. But, $min = max - \vartheta$ which means $\mu \geq \beta_h - \vartheta$ is always true in this case. Therefore, in

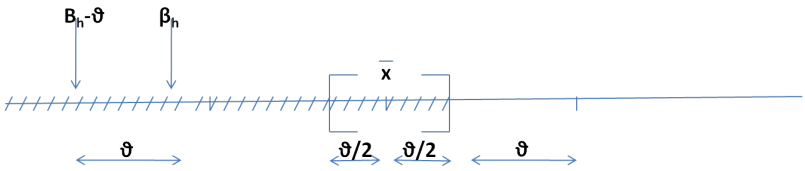

Figure 4.13: The "clear accept" case of the second method.

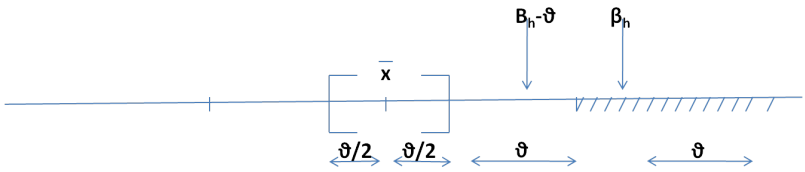

Figure 4.14: The "clear reject" case of the second method.

either case, we have $\mu \geq \beta_h - \vartheta$. Thus, with probability α_c, we can classify v as being (α_c,ϑ)-consistent in this case as well.

So, based on Observation 1 and Observation 2, when $\beta_h \leq max$ and the width of the CI is ϑ, with probability α_c we classify v as being (α_c,ϑ)-consistent and improving on the first approach. See Figure 4.13 for an illustration.

Theorem 1. *For any chance constraint h, for any assignment A in which $x = v$, for any ϑ, and for any α_c, for all values of $\beta_h \leq \bar{x} + \frac{\vartheta}{2}$, using a CI $[\bar{x} - \frac{\vartheta}{2}, \bar{x} + \frac{\vartheta}{2}]$ one can properly classify v as being (α_c,ϑ)-consistent with probability α_c or above.*

Proof. Follows immediately from Observation 1 and Observation 2.

Our third observation is as follows:

Observation 3. If $\beta_h > max + \vartheta$, then it is also true that $\beta_h - \vartheta > max$. Thus, since μ is covered by the CI $[\bar{x} - \frac{\vartheta}{2}, \bar{x} + \frac{\vartheta}{2}]$ with probability α_c and even in the worst case when $\mu = max$ we still have $\mu < \beta_h - \vartheta$, with probability α_c, we can classify v as being (α_c,ϑ)-inconsistent in this case. See Figure 4.14 for an illustration.

Theorem 2. *For any chance constraint h, for any assignment A in which $x = v$, for any ϑ, and for any α_c, for all values of $\beta_h > \bar{x} + \frac{3\vartheta}{2}$, using a CI $[\bar{x} - \frac{\vartheta}{2}, \bar{x} + \frac{\vartheta}{2}]$ one can properly classify v as being (α_c,ϑ)-inconsistent with probability α_c or above.*

Proof. Follows immediately from Observation 3.

Finally, we identify the case where we fail to make an inference with probability at least α_c:

Observation 4. If $max < \beta_h \leq max + \vartheta$, then $\beta_h - \vartheta \in]\bar{x} - \frac{\vartheta}{2}, \bar{x} + \frac{\vartheta}{2}]$.. Since, with coverage probability α_c, μ is covered by $[\bar{x} - \frac{\vartheta}{2}, \bar{x} + \frac{\vartheta}{2}]$, it is impossible in this case to

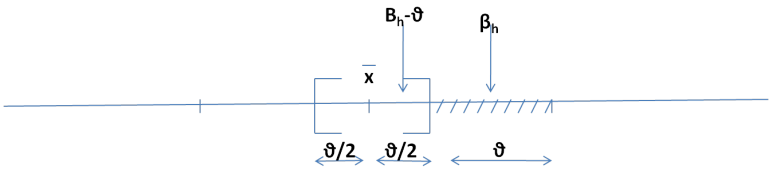

Figure 4.15: The critical case of the second method.

make a correct classification with probability α_c because we can either have $\mu \geq \beta\_\vartheta$ or $\mu < \beta_h - \vartheta$. So, this is the critical case in which we fail to make a correct inference with probability α_c or above. See Figure 4.15 for an illustration.

Therefore, one can show the following:

Theorem 3. *For any chance constraint h, for any assignment A in which $x = v$, for any ϑ, and for any α_c, there exists always a value for β_h such that it is impossible, using the CI approach, to properly classify v as being (α_c,ϑ)-consistent or not with probability α_c or above.*

Proof. Follows immediately from Observation 4 by setting $\beta_h \in]\bar{x} + \frac{\vartheta}{2}, \bar{x} + \frac{3\vartheta}{2}]$ $\square$.

In this critical case, one way is to decide which is the better situation: (1) classify v as being consistent while it may be inconsistent (increase the chances of type II error); or (2) classify v as being inconsistent while it may be consistent (increase the chances of type I error). Indeed, in SCSPs, it is better to reject a consistent value rather than accepting an inconsistent one in general.

Thus, our new inference method based on CIs in its two versions is as follows:

Step 1: Find a large enough sample size n and construct a confidence interval $[\bar{x} - \frac{\vartheta}{2}, \bar{x} + \frac{\vartheta}{2}]$ so that its coverage probability is α_c;

Step 2: We have three mutually exclusive cases:

Consistent: If $\beta_h \leq \bar{x} + \frac{\vartheta}{2}$, then classify v as (α_c,ϑ)-consistent;

Inconsistent: If $\beta_h > \bar{x} + \frac{3\vartheta}{2}$, then classify v as (α_c,ϑ)-inconsistent;

Mixed: If $\beta_h \in]\bar{x} + \frac{\vartheta}{2}, \bar{x} + \frac{3\vartheta}{2}]$, we select either of the two choices:

Version I: If you can tolerate more type II errors, then classify v as (α_c,ϑ)-consistent;

Version II: If you can tolerate more type I errors, then classify v as (α_c,ϑ)-inconsistent;

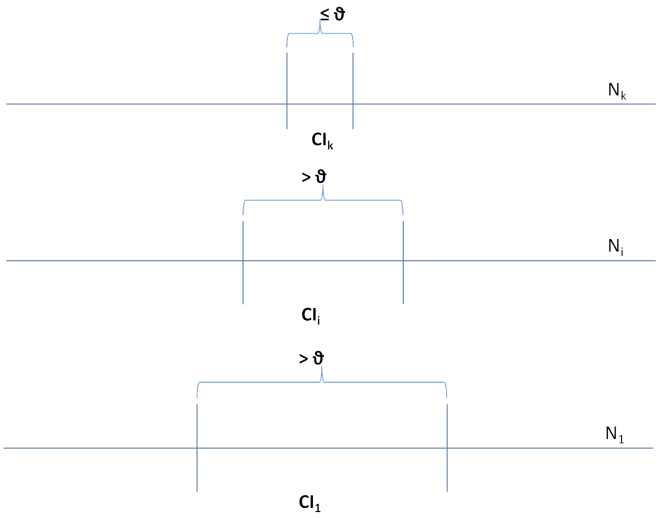

Figure 4.16: Dynamic computation of a large enough sample n to guarantee a CI of width ϑ or less.

$\vartheta = 0.01$	28000
$\vartheta = 0.009$	34000
$\vartheta = 0.008$	42500
$\vartheta = 0.007$	55500
$\vartheta = 0.005$	109000
$\vartheta = 0.001$	2750000

Figure 4.17: The initial theoretical sample size n needed as ϑ varies to guarantee a 0.95 confidence level as per Definition 2 in [RHTP15].

We are still left with how to compute a large enough sample size n to guarantee a CI of width ϑ or less. What we propose is to start with an initial sample size, say n_1 computed as per Definition 2 in [RHTP15]. If the CI's width is still larger than ϑ, we increase the sample size and keep doing so till we reach a sample size, say n_k, after k iterations till the width is ϑ or below. We illustrate this dynamic approach of computing a large enough sample size that produces a CI of width ϑ or less in Figure 4.16.

4.3.3 Validating the Second Approach

In Figure 4.17 we present the initial sample size we start with varying ϑ whereas in Figure 4.18, we present the one dynamically found that guarantees that the width of the confidence interval is smaller than or equal to ϑ. We notice that in practice, the

$\vartheta = 0.01$	39000
$\vartheta = 0.009$	48000
$\vartheta = 0.008$	61000
$\vartheta = 0.007$	79000
$\vartheta = 0.005$	155000
$\vartheta = 0.001$	3900000

Figure 4.18: The large enough sample size n needed as ϑ varies to guarantee a 0.95 confidence level computed dynamically.

	CCR: Version I	CCR: Version II	True classification
$\beta_h = 0.49$	**100%**	**100%**	Consistent
$\beta_h = 0.5$	**100%**	**98.3%**	Consistent
$\beta_h = 0.51$	**97.2%**	2.4%	Consistent (Mixed case)
$\beta_h = 0.52$	**97.8%**	**100%**	Inconsistent

Figure 4.19: The correct classification rate (CCR) for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.01$ using the second approach based on confidence intervals.

	CCR: Version I	CCR: Version II	True classification
$\beta_h = 0.49$	**100%**	**100%**	Consistent
$\beta_h = 0.5$	**100%**	**97.6%**	Consistent
$\beta_h = 0.51$	7%	**98.7%**	Inconsistent (Mixed case)
$\beta_h = 0.52$	**99%**	**100%**	Inconsistent

Figure 4.20: The correct classification rate (CCR) for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.009$ using the second approach based on confidence intervals.

sample size which is large enough to achieve the desired CI width is larger than the theoretical one found as per Definition 2 in [RHTP15].

In Figure 4.19, Figure 4.20, Figure 4.21, Figure 4.22, Figure 4.23, and Figure 4.24, we present the results of our experiments when $\alpha_c = 0.95$ for various values of μ, β_h, and ϑ. Each table shows the results of our second method in two versions: (1) when we are in the mixed case, we do classify v as consistent (referred to as Version I); (2) when we are in the mixed case, we classify v as inconsistent instead (referred to as Version II). When for a given β_h, and ϑ we are in the mixed case (i.e., $\beta_h \in]\bar{x} + \frac{\vartheta}{2}, \bar{x} + \frac{3\vartheta}{2}]$), we annotate that in the corresponding row in the table of results as "Mixed case".

- Figure 4.19, Figure 4.20, Figure 4.21, and Figure 4.22 do represent situations in

	CCR: Version I	CCR: Version II	True classification
$\beta_h = 0.49$	**100%**	**100%**	Consistent
$\beta_h = 0.5$	**100%**	**97.8%**	Consistent
$\beta_h = 0.51$	15.4%	**99.7%**	Inconsistent (Mixed case)
$\beta_h = 0.52$	**100%**	**100%**	Inconsistent

Figure 4.21: The correct classification rate (CCR) for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.008$ using the second approach based on confidence intervals.

	CCR: Version I	CCR: Version II	True classification
$\beta_h = 0.49$	**100%**	**100%**	Consistent
$\beta_h = 0.5$	**100%**	**97.6%**	Consistent
$\beta_h = 0.51$	38.1%	**99.8%**	Inconsistent (Mixed case)
$\beta_h = 0.52$	**100%**	**100%**	Inconsistent

Figure 4.22: The correct classification rate (CCR) for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.007$ using the second approach based on confidence intervals.

	CCR: Version I	CCR: Version II	True classification
$\beta_h = 0.49$	**100%**	**100%**	Consistent
$\beta_h = 0.5$	**100%**	**97.5%**	Consistent
$\beta_h = 0.51$	**97.8%**	**100%**	Inconsistent
$\beta_h = 0.52$	**100%**	**100%**	Inconsistent

Figure 4.23: The correct classification rate (CCR) for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.005$ using the second approach based on confidence intervals.

	CCR: Version I	CCR: Version II	True classification
$\beta_h = 0.49$	**100%**	**100%**	Consistent
$\beta_h = 0.5$	**100%**	**97.2%**	Consistent
$\beta_h = 0.51$	**100%**	**100%**	Inconsistent
$\beta_h = 0.52$	**100%**	**100%**	Inconsistent

Figure 4.24: The correct classification rate (CCR) for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.001$ using the second approach based on confidence intervals.

which we experience the three cases: "Consistent", "Inconsistent", and "Mixed". The results are inline with the theory and we do achieve a 95% or above CCR

in all cases of "Consistent" and "Inconsistent". Also, when we are in a "Mixed" case and the value is truly consistent, Version I as expected achieves a 95% or above CCR whereas Version II achieve a very low CCR. When, instead, we are in a "Mixed" case and the value is truly inconsistent, Version II as expected achieves a 95% or above CCR whereas Version I achieve a very low CCR.

- Figure 4.23 and Figure 4.24 do represent situations in which we experience only the two cases: "Consistent" and "Inconsistent". The results are inline with the theory and we do achieve a 95% or above CCR in all cases of "Consistent" and "Inconsistent".

- As the error threshold value ϑ gets smaller, the sample size required by our method and computed dynamically increases significantly.

In summary, the experimental results confirm that the second method (both under Version I or Version II) does achieve indeed a 95% CCR or above as expected in theory when we are outside the critical case, i.e., when $\beta_h \notin]\bar{x} + \frac{\vartheta}{2}, \bar{x} + \frac{3\vartheta}{2}]$. Furthermore, the width of the interval of the critical case of the second method is narrower than the width of the interval of the critical case of the first method. When $\beta_h \in]\bar{x} + \frac{\vartheta}{2}, \bar{x} + \frac{3\vartheta}{2}]$, then depending on whether v is truly consistent or not either the second method under Version I achieves at least a 95% CCR or Version II. Thus, we could safely state the superiority of the second method when compared to the first one which does more correct inference and does overcome the flaw of the first method when $\mu = \beta_h - \vartheta$.

4.4 Enforcing (α_c,ϑ)-Consistency

So far, we have restricted our analysis to just one assignment A and with respect to this assignment we looked at how to infer, with confidence α_c whether a value v is (α_c, ϑ)-consistent or not. In this section, we consider the question of how to make a chance constraint (α_c, ϑ)-GAC and also how to make a whole SCSP (α_c, ϑ)-GAC.

Now, the approaches we have presented in the previous sections treats each value v separately wrt to a given assignment A and is indeed able to detect whether each value is (α_c,ϑ)-consistent wrt a given assignment or not. But, if each value is (α_c,ϑ)-consistent, how about all values considered simultaneously in a chance constraint, in more than one chance constraint, and in the whole problem?

Recall that a chance constraint h is (α_c, ϑ)-GAC iff *simultaneously* every value in the domain of every variable in X_h is (α_c, ϑ)-GAC.

Let us consider the cross product of all the domains of the decision variable in X_h, i.e. our assignment space for chance constraint h denoted by $\mathcal{A}$.

Constraints:

$h_1 : pr\{C(x, y, r_1, r_2)\} \geq \beta_1$

$h_2 : pr\{C(y, z, r_1, r_2)\} \geq \beta_2$

$h_3 : pr\{C(x, z, r_1, r_2)\} \geq \beta_3$

Decision variables:

$x, y, z \in \{0, 1\}$

Random variables:

r_1, r_2: has infinite support

Stage structure:

$V_1 = \{x, y, z\}$

$S_1 = \{r_1, r_2\}$

$L = [\langle V_1, S_1 \rangle]$

Figure 4.25: A single-stage SCSP.

Using, the approach outlined in the previous section, for each value v in the domain of every decision variable $x \in X_h$ one of these two outcomes is possible:

- There exists an assignment $A \in \mathcal{A}$ for which value v is (α_c, ϑ)-consistent;
- For every assignment $A \in \mathcal{A}$, value v is (α_c, ϑ)-inconsistent.

A naive approach is to simply consider each value separately and if it is (α_c, ϑ)-inconsistent, we prune it. Otherwise, we leave it in the domain. But, the problem is that errors will accumulate and overall we may not achieve a confidence level of α_c.

Let us illustrate this situation by the following experiment. Consider the single-stage SCSP shown in Figure 4.25. We have three binary decision variables and three chance constraints as depicted in Figure 4.26. We have two random variables r_1 and r_2 involved in each chance constraint and have an infinite support.

Assume that $\beta_1 = \beta_2 = \beta_3 = 0.5$, $\vartheta = 0.01$, and $\alpha_c = 0.95$. Assume further that for each chance constraint, we know the true mean for each assignment and hence we are able to precisely know whether a certain assignment satisfies the chance constraint or not. In Figure 4.27, Figure 4.28, and Figure 4.29, we show the consistent and inconsistent assignments for chance constraint h_1, h_2, h_3, respectively.

Based on Figure 4.27, value 0 in the domain of x and value 1 in the domain of y are truly consistent, but value 1 in the domain of x and value 0 in the domain of y are truly inconsistent.

Based on Figure 4.28, value 1 in the domain of y and value 0 in the domain of z

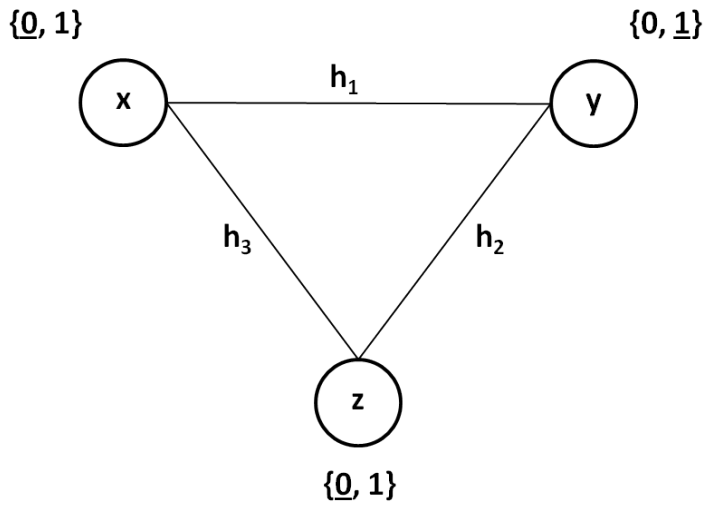

Figure 4.26: Constraint Network of SCSP in Figure 4.25.

Assignment	True mean	True classification
$A_1 : \langle x = 0, y = 0 \rangle$	$pr\{C(x = 0, y = 0, r_1, r_2)\} = 0.48$	Inconsistent
$A_2 : \langle x = 0, y = 1 \rangle$	$pr\{C(x = 0, y = 1, r_1, r_2)\} = 0.5$	Consistent
$A_3 : \langle x = 1, y = 0 \rangle$	$pr\{C(x = 1, y = 0, r_1, r_2)\} = 0.42$	Inconsistent
$A_4 : \langle x = 1, y = 1 \rangle$	$pr\{C(x = 1, y = 0, r_1, r_2)\} = 0.4$	Inconsistent

Figure 4.27: Consistent and inconsistent assignments in h_1 wrt to $\beta_1 = 0.5$ and $\vartheta = 0.01$.

Assignment	True mean	True classification
$A_5 : \langle y = 0, z = 0 \rangle$	$pr\{C(y = 0, z = 0, r_1, r_2)\} = 0.42$	Inconsistent
$A_6 : \langle y = 0, z = 1 \rangle$	$pr\{C(y = 0, z = 1, r_1, r_2)\} = 0.48$	Inconsistent
$A_7 : \langle y = 1, z = 0 \rangle$	$pr\{C(y = 1, z = 0, r_1, r_2)\} = 0.49$	Consistent
$A_8 : \langle y = 1, z = 1 \rangle$	$pr\{C(y = 1, z = 0, r_1, r_2)\} = 0.4$	Inconsistent

Figure 4.28: Consistent and inconsistent assignments in h_2 wrt to $\beta_2 = 0.5$ and $\vartheta = 0.01$.

are truly consistent, but value 0 in the domain of y and value 1 in the domain of z are truly inconsistent.

Based on Figure 4.29, value 0 in the domain of x and value 0 in the domain of z are truly consistent, but value 1 in the domain of x and value 1 in the domain of z are truly inconsistent.

Therefore, an algorithm that enforces (α_c,ϑ)-consistency should *simultaneously* prune 1 from the domain of x, value 0 from the domain of y, and value 1 from the domain of

Assignment	True mean	True classification
$A_9 : \langle x = 0, z = 0 \rangle$	$pr\{C(x = 0, z = 0, r_1, r_2)\} = 0.6$	Consistent
$A_{10} : \langle x = 0, z = 1 \rangle$	$pr\{C(x = 0, z = 1, r_1, r_2)\} = 0.48$	Inconsistent
$A_{11} : \langle x = 1, z = 0 \rangle$	$pr\{C(x = 1, z = 0, r_1, r_2)\} = 0.42$	Inconsistent
$A_{12} : \langle x = 1, z = 1 \rangle$	$pr\{C(x = 1, z = 0, r_1, r_2)\} = 0.4$	Inconsistent

Figure 4.29: Consistent and Inconsistent Assignments in h_3 wrt to $\beta_1 = 0.5$ and $\vartheta = 0.01$.

z in $\alpha_c\%$ of the times while allowing an error of ϑ.

But, what if we apply our naive approach, what is the confidence level achieved for one chance constraint (namely h_1), two chance constraints (namely h_1 and h_2 simultaneously), and the whole problem in Figure 4.25 (namely h_1, h_2, and h_3 simultaneously) if for each value independently we enforce (α_c,ϑ)-consistency using the approach in the previous section.

Our experimental setup is as follows. We run 1000 experiments where each experiment is as follows:

1. For each assignment $A_i \in \{A_1, \ldots, A_{12}\}$, since $\alpha_c = 0.95$ and $\vartheta = 0.01$, we generate an independent sample of size $n = 39000$[1].

2. For each value v in the domain of every decision variable (x, y, and z), if, for any assignment A_i in which v appears, our method in the previous section classifies v as (α_c,ϑ)-inconsistent, we prune v from the domain of the decision variable.

3. The overall number of correct classification is updated as follows:

   Correct classification wrt to h_1: we increment the number of correct classification wrt to h_1 by one if we simultaneously prune value 1 from the domain of x and value 0 from the domain of y.

   Correct classification wrt to h_1 and h_2: we increment the number of correct classification wrt to h_1 and h_2 by one if we simultaneously prune value 1 from the domain of x, value 0 from the domain of y, and value 1 from the domain of z.

   Correct classification wrt to h_1, h_2, and h_3: we increment the number of correct classification wrt to h_1, h_2, and h_3 by one if we simultaneously prune value 1 from the domain of x, value 0 from the domain of y, and value 1

[1]Determined from the previous experiments

from the domain of z. Note that this case is different from the previous case because in the previous case, only assignments $A-1$ through A_8 are used to make the inference whereas in this case all assignments are used in the inference process.

Finally, the CCR wrt to h_1 is the total number of correct classification wrt to h_1 divided by 1000. Similarly, we compute the CCR wrt h_1 and h_2 and the CCR wrt h_1, h_2, and h_3.

The results of the experiments are shown in Table 4.30.

CCR wrt to h_1	CCR wrt h_1 and h_2	CCR wrt h_1, h_2, and h_3
97.1%	91.4%	88.8%

Figure 4.30: CCR for multiple chance constraints.

As expected, as the number of multiple inferences increase, the overall confidence level drops as errors accumulate. So, how to cure this problem?

As $|\mathcal{A}|$ increase, so does the probability of misclassifying values. Thus, in practice, to enforce that a single chance constraint h is (α_c^h,ϑ)-GAC that is composed of m assignments, each value needs to be (α_c,ϑ)-consistent wrt to one assignment where α_c can adjusted using the conservative Bonferoni correction [GB97]:

$$(1-\alpha_c^h) = \sum_{i=1}^{m}(1-\alpha_c)$$

So in order to make h_1 $(\alpha_c^h,\ \vartheta)$-GAC, where $\alpha_c^h = 0.8$, the adjusted α_c should be 0.95 since there are 4 assignments and

$$(1-0.8) = 0.2 = \sum_{i=1}^{4}(1-0.95) = \sum_{i=1}^{4}(0.05)$$

To make a whole SCSP $(\alpha_c^g,\ \vartheta)$-GAC composed of n chance constraints, the previous approach of Bonferroni correction can be extended to achieve this. The correction is as follows:

$$(1-\alpha_c^g) = \sum_{i=1}^{nm}(1-\alpha_c)$$

So in order to make h_1 and h_2 simultaneously $(\alpha_c^g,\ \vartheta)$-GAC, where $\alpha_c^g = 0.6$, the adjusted α_c should be 0.95 since there are 8 assignments and

$$(1-0.6) = 0.4 = \sum_{i=1}^{8}(1-0.95) = \sum_{i=1}^{8}(0.05)$$

Similarity, in order to make the whole SCSP in Figure 4.26 simultaneously (α_c^g, ϑ)-GAC, where $\alpha_c^g = 0.4$, the adjusted α_c should be 0.95 since there are 12 assignments and

$$(1 - 0.4) = 0.6 = \sum_{i=1}^{12}(1 - 0.95) = \sum_{i=1}^{12}(0.05)$$

The Bonferroni correction is indeed very conservative as shown by our experiments. In table 4.31 we show the expected CCR, in theory and in practice, after we make a Bonferroni correction for a single chance constraint h_1, two chance constraints h_1 and h_2 simultaneously, and three chance constraints h_1, h_2, and h_3 simultaneously.

	CCR wrt to h_1	CCR wrt h_1 and h_2	CCR wrt h_1, h_2, and h_3
Theory	80%	60%	40%
Practice	97.1%	91.4%	88.8%

Figure 4.31: CCR for multiple chance constraints in theory vs. practice.

The good news is that our methods in practice achieves much higher CCR than expected. But, it seems that less conservative approaches need to be further studied and explored in the future. Indeed, if we wish to achieve an overall confidence level of 0.95, the adjusted individual confidence level with 10 assignments, using the Bonferroni correction, would be 0.995 which is too high which would make the Bonferroni correction too demanding.

As an alternative solution that would improve the overall confidence level by reducing the individual errors further is to use a larger sample size than the one found by our method. By increasing the sample size, we reduce the errors and hence improve the overall confidence level. Furthermore, since there is very little cost in using a higher sample size in our method, it is still practical to do so. For instance, in the previous experiments where for each value we maintain (α_c,ϑ)-consistency where $\alpha_c = 0.95$ and $\vartheta = 0.01$, the sample size found and used by our method was $n = 39000$ but if we increase it to 50000, we get the results shown in table 4.32 in which, in all cases, an overall confidence level of α_c is achieved.

	CCR wrt to h_1	CCR wrt h_1 and h_2	CCR wrt h_1, h_2, and h_3
Practice ($n = 39000$)	97.1%	91.4%	88.8%
Practice ($n = 50000$)	99.6%	96.1%	95.7%

Figure 4.32: CCR for multiple chance constraints in practice with increased sample size.

4.5 Conclusion

In this Chapter, we have proposed two methods for enforcing (α_c,ϑ)-consistency. We presented and experimentally validated a first approach based on confidence intervals to infer whether or not a value v is (α_c,ϑ)-consistent with respect to an assignment A. The empirical results confirm the expected theoretical results which state that a CCR of at least $\alpha_c 100\%$ can be achieved as long as $\mu \notin]\beta_h - \vartheta, \beta_h + \vartheta[$. Otherwise, the proposed method does not have any guarantees about the CCR that can be achieved. Furthermore, in the case when $\mu = \beta_h - \vartheta$, the first approach is flawed and always makes the wrong influence.

Next, based on an analysis of the first approach by taking a different perspective which focuses on building a confidence interval around the sample mean $\bar{x}$ rather than around β_h as in the first approach, we introduced our second statistical inference method. We proposed two versions of the second method to deal with the critical case when we are unable to guarantee a CCR of at least 95%: (1) a first version in which we tolerate the error of misclassifying an inconsistent value as being consistent; and (2) a second version in which we tolerate the error of misclassifying a consistent value as being inconsistent instead. The second method (in both versions) not only improves on the cases in which we are able to make more correct inference at a CCR of at least $\alpha_c 100\%$, it also remedies the flaw when $\mu = \beta_h - \vartheta$ when compared to the first approach. Our experiments also confirm and validate our theoretical properties of the second method in both of its versions and show the superiority of the second method when compared to the first one.

Finally, we have shown how to enforce (α_c,ϑ)-GAC for a chance constraint and how to make a SCSP (α_c,ϑ)-GAC by using a Bonferroni correction approach of the confidence probability α_c.

Chapter 5

Enforcing Statistical Consistency via Composite Hypothesis Testing

In this chapter, when the sample size is large enough, we assume instead of being the probability of success of a Bernoulli distribution, the satisfaction probability can be seen as the mean of a Normal distribution. While the use of the Normal distribution seems odd at first, it is supported by the central limit theorem and with sufficiently large n, the Normal distribution is a good estimate of the Binomial distribution. Indeed, according to [BCNN00, MR03], when the sample size n is 50 or more, the sampling distribution can be assumed to be the Normal distribution. The main reason why we make this assumption is to be able to tackle the problem using a hypothesis testing approach. The rest of the chapter is organised as follows. In Section 5.1, we review hypothesis testing. Then, in Section 5.2, we show how to classify if a value v is (α_c,ϑ)-consistent or not by means of composite hypothesis testing. Then, we validate our approach in Section 5.3 and show the superiority of the new approach compared to the previous ones based on confidence intervals. Finally, we conclude in Section 5.4.

5.1 Hypothesis Testing

A *statistical hypothesis* is an assumption about a population parameter. This assumption may or may not be true. *Hypothesis testing* refers to the formal procedures used by statisticians to accept or reject statistical hypotheses. The methodology employed by the analyst depends on the nature of the data used, as well as the goals of the analysis. The goal is either to accept or to reject the null hypothesis.

The best way to determine whether a statistical hypothesis is true would be to examine the entire population. Since that is often impractical, statisticians typically examine a random sample from the population. If the sample data are not consistent

with the statistical hypothesis, the hypothesis is rejected. There are two types of statistical hypotheses:

Null hypothesis: This refers to any hypothesis we wish to test and is denoted by H_0. It is usually the hypothesis that sample observations result purely from chance. The rejection of H_0 leads to the acceptance of an alternative hypothesis. H_0 is a hypothesis which the researcher tries to disprove, reject, or nullify.

Alternative hypothesis: This refers to hypothesis that sample observations are influenced by some non-random cause and is denoted by H_1 or H_a.

The null hypothesis refers to the common view of something while the alternative hypothesis is what the research really thinks is the cause of the phenomenon. For example, suppose we want to determine whether a coin was fair and balanced. A null hypothesis might be that half of the flips would result in heads and half in tails. The alternative hypothesis might be that the number of heads and tails would be very different.

Symbolically, these hypotheses would expressed as

$$H_0 : \ p = 0.5$$

$$H_a : \ p \neq 0.5$$

Suppose we flipped the coin 50 times, resulting in 40 heads and 10 tails. Given this result, we would reject the null hypothesis and we would conclude, based on the evidence, that the coin was probably not fair and balanced.

The terms of acceptance and rejection refer only to our decision based on incomplete information and we should know that we could be wrong. Since, we will have taken action based on our estimate about the population, the acceptance of a hypothesis implies that data do not give sufficient evidence to refute it. Rejection implies that the sample evidence refutes it.

Based on sample data, statisticians follow a formal process to determine whether to reject a null hypothesis. This process of hypothesis testing consists of four basic steps:

1. State the hypotheses. This means stating the null and alternative hypotheses which are mutually exclusive. That is, if one is true, the other must be false.

2. Formulate an analysis plan which describes how to use sample data to evaluate the null hypothesis. The evaluation often focuses around a single test statistic.

	H_0 is true	H_0 is false
Accept H_0	correct decision	Type II error
Reject H_0	Type I error	correct decision

Figure 5.1: Four possible outcomes of hypotheses testing.

3. Analyse sample data. Find the value of the test statistic (mean score, proportion, t-score, z-score, etc) described in the analysis plan. The test statistic is computed as follows:

$$Test\ statistic = \frac{relevant\ statistic - hypothesized\ parameter}{standard\ error\ of\ relevant\ statistic}$$

For example the value of the z-score is computed as follows:

$$z = \frac{\bar{x} - \mu_0}{\frac{\sigma}{\sqrt{n}}}$$

4. Interpret the results. Apply the decision rule described in the analysis plan. If the value of the test statistic is unlikely based on the null hypothesis, reject the null hypothesis.

Two types of errors can result from a hypothesis test:

Type I error: it occurs when the researcher rejects a null hypothesis when it is true. The probability of committing a type I error is called the *significance level*. This probability is also called *alpha* and is denoted by α.

Type II error: it occurs when the researcher fails to reject a null hypothesis that is false. The probability of committing a type II error is called *beta* and is often denoted by β. The probability of not committing a type II error is called the *power of the test*. It is the probability of rejecting H_0 given that a specific alternative hypothesis is true which is $1 - \beta$.

In testing any statistical hypothesis, there are four possible outcomes that determine whether our decision is correct or in error. These four situations are summarized in Fig. 5.1

The analysis plan includes decision rules for rejecting the null hypothesis. These decision rules can be described in two ways, with reference to a region of acceptance or with reference to a p-value.

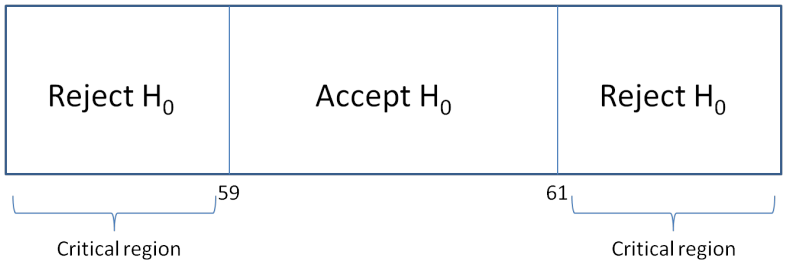

Figure 5.2: Decision criterion.

The *region of acceptance* is defined so that the chance of making a type I error is equal to the significance level. The region of acceptance is a range of values. If the test statistic falls within the region of acceptance, the null hypothesis is not rejected. The set of values outside the region of acceptance is called the *region of rejection*. If the test statistic falls within the region of rejection, the null hypothesis is rejected. In such cases, we say that the hypothesis has been rejected at the α level of significance.

The *p-value* is the lowest level (of significance) at which the observed value of the test statistic is significant. The p-value or calculated probability is the estimated probability of rejecting the null hypothesis H_0 of a study question when the null hypothesis is true. In other words, the p-value may be considered as the probability of obtaining a result at least as extreme as the one that was actually observed, assuming that the null hypothesis is true. Indeed, the smaller the p-value, the greater the evidence is against the null hypothesis. If we have a given significance level α then we reject H_0 if the p-value is less than or equal to α.

Example. Consider the Null hypothesis that the average weight of female students in a certain school is 60 kilograms against the alternative hypothesis that is unequal to 60. That is, we wish to test:

$$H_0 : \mu = 60$$

$$H_1 : \mu \neq 60$$

The alternative hypothesis means that $\mu < 60$ or $\mu > 60$. In this case, the sample mean is the test statistic. The critical region for the test statistic may be the two intervals $\bar{x} < 59$ and $\bar{x} > 61$. The acceptance region will then be the interval $59 \leq \bar{x} \leq 61$. This decision criterion is illustrated in Figure 5.2.

The probability of committing a type I error (the level of significance of our test α) is equal to the sum of the areas that have been shaded in each tail of the distribution in Figure 5.3.

Therefore,

$$\alpha = P(\bar{x} < 59 \; when \; \mu = 60) + P(\bar{x} > 61 \; when \; \mu = 60)$$

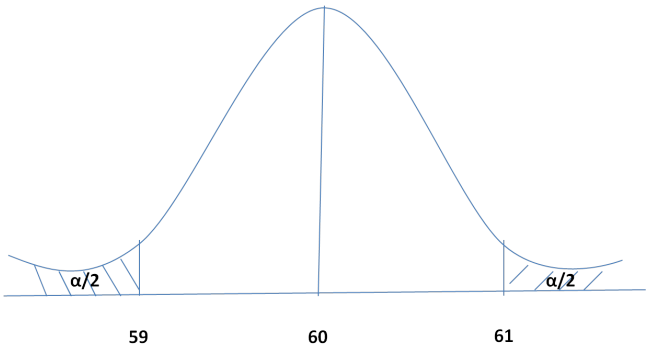

Figure 5.3: Critical region for testing $\mu = 60$ versus $\mu \neq 60$.

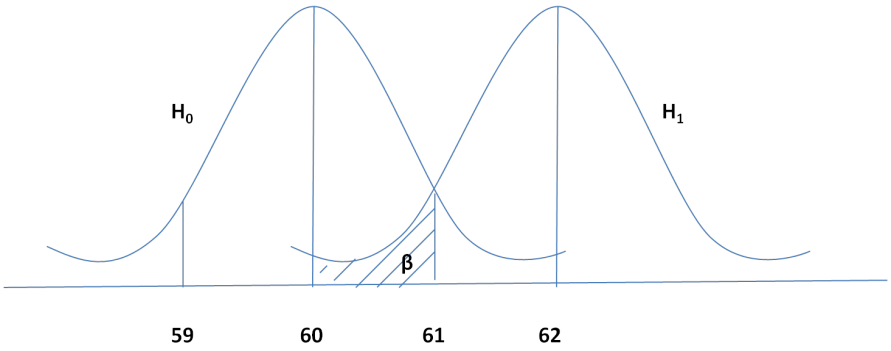

Figure 5.4: Type II error for testing $\mu = 60$ versus $\mu \neq 60$.

A type II error will result when the sample mean $\bar{x}$ falls between 59 and 61, when H_1 is true. Therefore, by referring to Figure 5.4 we find that:

$$\beta = P(59 \leq \bar{x} \leq 61, \ when \ \mu = 62 \ (for \ example))$$

There are important properties of a test of a hypothesis:

1. type I and type II errors are related in such a way that an increase in the probability of one will cause a decrease in the probability of the other in general.

2. The size of the critical region, and hence the probability of committing a type I error, can always be reduced by adjusting the critical value(s).

3. an increase in the sample size will decrease α and β simultaneously.

4. If the Null hypothesis is false, β is maximum when the true value of a parameter approaches the hypothesized value. The greater the distance between the true value and the hypothesized value, the smaller β will be.

A test of a statistical hypothesis, when the region of rejection is on only one side of the sampling distribution, is called a *one-tailed test*. For example, suppose that the null hypothesis states that the mean is equal to 10. The alternative hypothesis is that the mean is greater than 10. The region of rejection would consist of a range of values located on the right side of the sampling distribution. That is, a set of values greater than 10.

A test of a statistical hypothesis, where the region of rejection is on both sides of the sampling distribution, is called a *two-tailed test*. For example, suppose the null hypothesis states that the mean is equal to 10. The alternative hypothesis would be that the mean is different from 10. The region of rejection would consist of a range of values located on both sides of the sampling distribution, i.e., the region of rejection would consist partly of values that were less than 10 and partly of values greater than 10.

A hypothesis is called *composite* when the parameter space ϑ is divided into two disjoint regions, ϑ_0 and ϑ_1. The test is written as follows:

$$H_0 : \ \vartheta \in \vartheta_0$$

$$H_1 : \ \vartheta \notin \vartheta_1$$

Rejection and failure to reject the null hypothesis, critical regions, and type I and II errors have the same meaning for a composite hypothesis as it does with a simple hypothesis.

5.2 Statistical Inference Rules Through Hypothesis Testing

This section shows how to formulate the problem of classifying values as being (α_c,ϑ)-consistent as a composite hypothesis testing problem. In subsection 5.2.1, we describe our Null and alternative hypothesis. In subsection 5.2.2, we show how to calculate the sample size that is large enough to guarantee a certain level of errors. Next, in subsection 5.2.3, we show the steps that allow us to classify whether or not a specific value v is (α_c,ϑ)-consistent. For the sake of clarity, we recall the following notation:

α_c	Confidence level
ϑ	Error threshold tolerance
β_h	Chance constraint probability satisfaction threshold
α	Significance level or type I error rate. α is $1 - \alpha_c$ throughout this thesis
β	Type II error rate
n	sample size
S_n	A random sample of size n

5.2.1 Hypothesis Formulation

We show how to enforce (α_c, ϑ)-consistency by means of hypothesis testing.

Recall that a value v in the domain of x is (α_c, ϑ)-consistent iff, with confidence α_c, there exists an assignment A in which $x = v$ and $\sum_{s \in \Omega_h} B_h^s pr(s) \geq \beta_h - \vartheta$.

Let $\mathcal{P}$ denote the following infinite population for a given assignment A in which $x = v$:

$$\mathcal{P} = \{B_h^s pr(s) | s \in \Omega_h\}$$

Indeed the expression $\sum_{s \in \Omega_h} B_h^s pr(s)$ can be seen as the mean, denoted by μ, of the entire population $\mathcal{P}$. It is impossible to evaluate μ exactly in general since $\mathcal{P}$ is infinite. Instead of considering the entire pool, one can draw a statistical sample. The information obtained from the sample would allow us to develop hypotheses about the larger infinite population $\mathcal{P}$.

Therefore, for a given assignment A in which $x = v$, we propose to draw a finite subset of *random samples* from $\mathcal{P}$ and formulate proper statistical hypothesis which would allow us, with confidence α_c, to make a statistical inference whether $\mu \geq \beta_h - \vartheta$ is true or not based on the evidence provided by the samples.

A possible hypothesis from which one can infer, with confidence α_c, whether or not v is (α_c, ϑ)-consistent can be stated as follows:

$$H_0 : \ \mu \geq \beta_h - \vartheta$$

$$H_1 : \ \mu < \beta_h - \vartheta$$

If, the evidence supports accepting the Null hypothesis, with confidence α_c, we simply infer that v is (α_c, ϑ)-consistent since in that case $\mu \geq \beta_h - \vartheta$ with confidence α_c. Otherwise, the Null hypothesis is rejected and v is inferred, with confidence α_c as being (α_c, ϑ)-inconsistent.

The choice of Null hypothesis is made in such a way that we wish to reject that $\mu \geq \beta_h - \vartheta$ since in most cases, most values are inconsistent and the alternative hypothesis

is often true. So, when we accept the Null hypothesis, based on the evidence with confidence α_c, then we are making sure that it is not purely from chance.

Note that both our Null hypothesis $H_0 : \ \mu \geq \beta_h - \vartheta$ and alternative hypothesis $H_1 : \ \mu < \beta_h - \vartheta$ are indeed composite hypotheses. So, let us shed some light on the similarities and the difference between simple Null hypothesis testing and composite Null hypothesis testing. In what follows, let $\mu_0 = \beta_h - \vartheta$.

In [LS99], the authors explain that the simple Null hypothesis approach and the composite Null hypothesis approach for one-tailed hypothesis tests are both valid. In the case of a simple Null hypothesis approach, the choices for a one-tailed test of a population mean μ with a specific numeric value μ_0 are either

$$H_0 : \ \mu = \mu_0$$

$$H_1 : \ \mu < \mu_0$$

or

$$H_0 : \ \mu = \mu_0$$

$$H_1 : \ \mu > \mu_0$$

Regardless of the form of the alternative hypothesis H_1 in this case, the Null hypothesis specifies the same value, μ_0. In the case of composite Null hypothesis, the choices are:

$$H_0 : \ \mu \geq \mu_0$$

$$H_1 : \ \mu < \mu_0$$

or

$$H_0 : \ \mu \leq \mu_0$$

$$H_1 : \ \mu > \mu_0$$

With this approach, the composite Null hypothesis specifies a range of possible values for the population mean depending upon the appropriate alternative hypothesis. Because of this difference in the form of the Null hypothesis, the two approaches require different statistical explanations. For example, let us consider the one-tailed hypothesis test about the mean of the population:

$$H_0 : \ \mu = \mu_0$$

$$H_1 : \ \mu < \mu_0$$

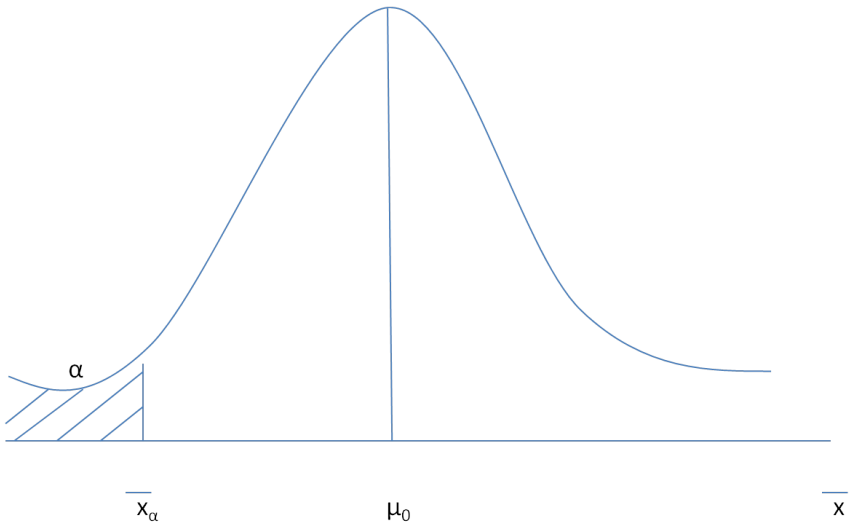

Figure 5.5: Sampling distribution of $\bar{x}$ under the simple Null hypothesis

with the simple Null hypothesis, the distribution of the sample mean $\bar{x}$ under the Null hypothesis is uniquely determined because the value of the population mean is uniquely specified, $H_0 : \mu = \mu_0$ (see Figure 5.5) and hence the probability of a type I error is uniquely determined by any given rejection region. Therefore, the appropriate rejection region of the test is the one for which the probability of its type I errors equals α, the chosen significance level (as shown in Figure 5.5).

However, using the composite Null hypothesis approach, the sampling distribution of $\bar{x}$ under the Null hypothesis is not uniquely determined because the value of the population mean is not uniquely specified. Since, under the composite Null hypothesis $H_0 : \mu \geq \mu_0$ (for example), there are infinite number of alternative sampling distributions some of which are shown in Figure 5.6.

Because of this multiplicity of sampling distributions, the probability of a type I error associated with any specific rejection region is not unique. However, the *largest* probability of a type I error for any chosen rejection region is associated with the sampling distribution centered at μ_0. We thus select the rejection region that makes the probability of type I error equal to α for a specific sampling distribution (but less than α for all other ones specified under the Null hypothesis).

Under this approach, the rejection region is chosen to make the *maximum* (but not the actual) probability of type I error equal to α (see Figure 5.6). For all these reasons, both approaches are statistically valid, yield identical rejection regions, and produce identical decisions. But, what really differs between the two approaches is the statistical explanation. The differences can be summarised as follows:

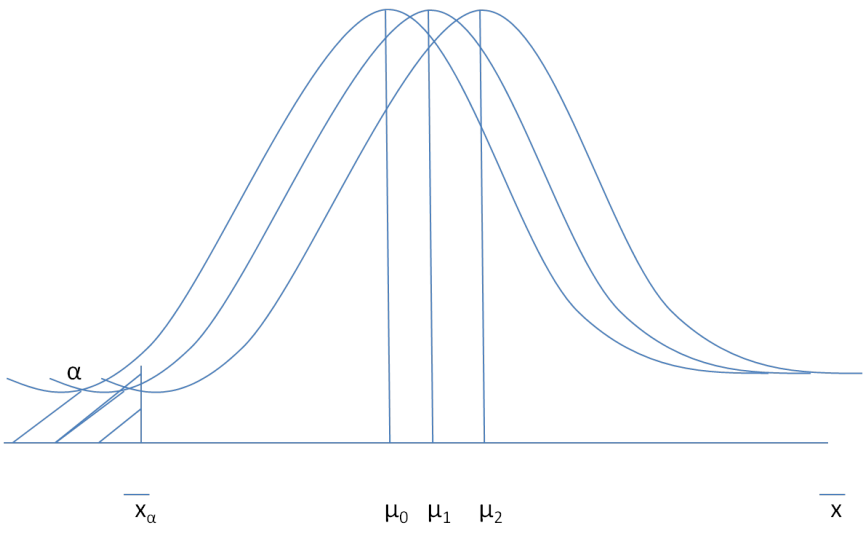

Figure 5.6: Sampling distribution of $\bar{x}$ under the composite Null hypothesis.

1. The existence of a single versus multiple sampling distributions under the Null hypothesis; and

2. The exact versus the conservative maximum probability of type I error associated with the level of significance, α.

5.2.2 Choice of sample size

The sample size is an important feature of an empirical study in which the goal is to make inferences about a population from a sample. The sample size should be chosen in such a way to achieve a good power for a fixed α and a specific alternative hypothesis. Suppose we want to test the hypothesis:

$$H_0 : \ \mu = \mu_0$$
$$H_1 : \ \mu > \mu_0$$

with a significance level α when the variance σ^2 is known. For a specific alternative $\mu = \mu_0 + \delta$[1], the power:

$$1 - \beta = P(\bar{x} > a \ when \ \mu = \mu_0 + \delta)$$

where a is the critical bound which is a value that bounds the critical region. The power of our test can be shown in Figure 5.7

[1] δ is the difference between the hypothesized mean and the true mean

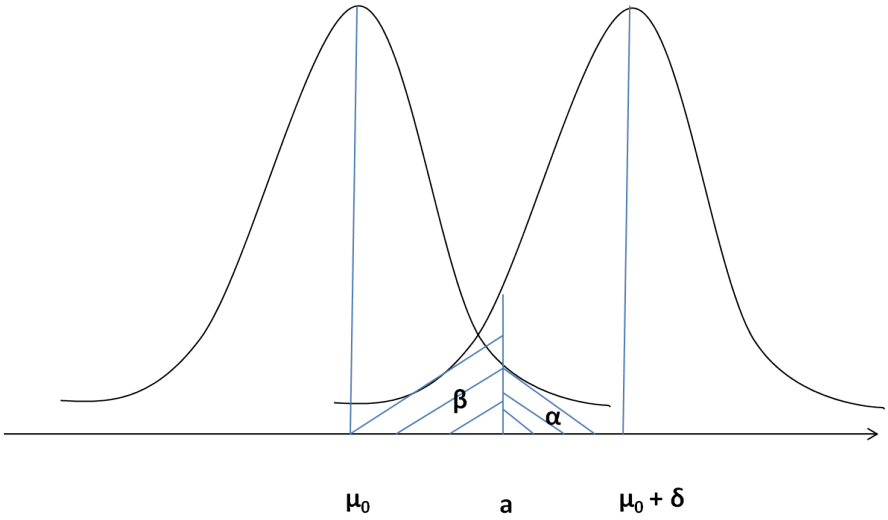

Figure 5.7: Testing $\mu = \mu_0$ versus $\mu = \mu_0 + \delta$.

Therefore,

$$\beta = P(\bar{x} \leq a \ when \ \mu = \mu_0 + \delta)$$

$$= P(\frac{\bar{x} - (\mu_0 + \delta)}{\frac{\sigma}{\sqrt{n}}} \leq \frac{a - (\mu_0 + \delta)}{\frac{\sigma}{\sqrt{n}}} \ when \ \mu = \mu_0 + \delta)$$

The statistic $\frac{\bar{x}-(\mu_0+\delta)}{\frac{\sigma}{\sqrt{n}}}$ is the standard Normal variable Z. Therefore,

$$\beta = P(Z < \frac{a - \mu_0}{\frac{\sigma}{\sqrt{n}}} - \frac{\delta}{\frac{\sigma}{\sqrt{n}}}) = P(Z < Z_\alpha - \frac{\delta}{\frac{\sigma}{\sqrt{n}}})$$

From which we conclude that

$$-Z_\beta = Z_\alpha - \frac{\delta\sqrt{n}}{\sigma}$$

and hence

$$n = \frac{(Z_\alpha + Z_\beta)^2\sigma^2}{\delta^2}$$

This result also holds true when H_1 is $\mu < \mu_0$ and in the case of a two-tailed test, we have:

$$n = \frac{(Z_{\frac{\alpha}{2}} + Z_\beta)^2\sigma^2}{\delta^2} \tag{5.1}$$

Note that when the σ is unknown, some statistics textbooks (e.g., [BCNN00]and [MR03]) suggest that one can safely replace σ by s (the sample standard deviation) in the test statistic $\frac{\bar{x}-(\mu_0+\delta)}{\frac{\sigma}{\sqrt{n}}}$ when $n \geq 30$ or $n \geq 50$ and still use the Z-tables for the appropriate critical region.

5.2.3 Value consistency classification via composite hypothesis testing

Given our composite Null hypothesis $H_0 : \mu \geq \beta_h - \vartheta$, we show, for an assignment A in which $x = v$, the steps that would allow us to classify whether a specific value v is (α_c,ϑ)-consistent or not.

Sampling: We first determine the appropriate sample size $n \geq 50$. Then, we use Simple Random Sampling, to generate a sample. Finally, we compute the sample mean $\bar{x}$ and the sample standard deviation s.

Computation of test statistic: The standard deviation σ of the population under study is indeed unknown. But, since $n \geq 50$, one can use the Z-score test statistic [MR03] denoted as $Z = \frac{\bar{x}-\mu}{SE}$ where SE denotes the standard error $SE = \frac{s}{\sqrt{n}}$.

Interpretation: We opt for the traditional method that finds the critical value cv. If the test statistic z is larger than or equal to the critical value cv, then we fail to reject H_0 (or: we accept H_0). Else, if $z < cv$, then we reject H_0 and accept H_1 instead.

Inference: If H_0 is not rejected, then the value v is classified as (α_c,ϑ)-consistent. Else, the value v is classified as (α_c,ϑ)-inconsistent.

Example. Suppose we are given a chance constraint $h : pr\{C\} >= \beta_h$ over an infinite number of scenarios where β_h is 0.76. Suppose we are given an assignment A in which $x = v$. Suppose we take a random sample of 36 scenarios and compute the satisfaction probability of $pr\{C\}$ for assignment A (i.e, the sample mean $\bar{x}$) and it happens to be 0.745 with a sample standard deviation equal to 0.06. Is the value v (α_c,ϑ)-consistent for $\alpha_c = 0.95$ and $\vartheta = 0.01$ or not?

Using our approach, we formulate the following hypothesis:

$$H_0 : \mu \geq \beta_h - \vartheta = 0.75$$

$$H_1 : \mu < \beta_h - \vartheta = 0.75$$

At $\alpha = 0.05$, we have $Z_0 = -1.645$ (see Figure 5.8) and

$$Z_0 = \frac{\bar{x} - 0.75}{\frac{\sigma}{\sqrt{n}}}$$

Thus,

$$\bar{x} = 0.75 + \frac{0.06}{\sqrt{36}}. - 1.645 = 0.7335$$

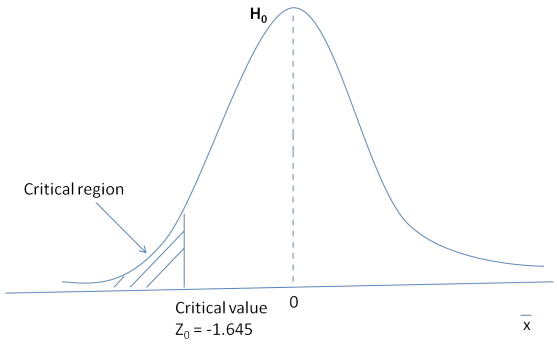

Figure 5.8: Critical region for testing $\mu \geq 0.75$ versus $\mu < 0.75$.

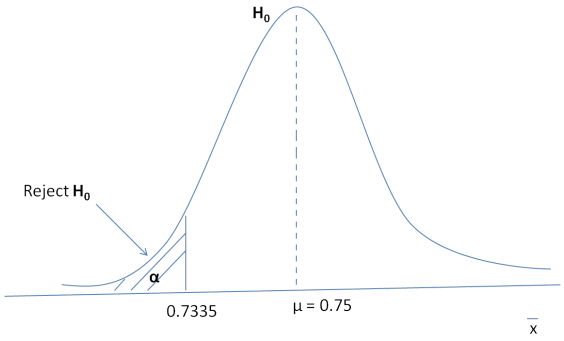

Figure 5.9: Probability of type I error for testing $\mu \geq 0.75$ versus $\mu < 0.75$.

The probability of committing a type I error or the level of significance of our test, is equal to the area of the shaded region to the left of $\bar{x} = 0.7335$ shown in Figure 5.9 ($P(Reject\ H_0|H_0\ is\ true)$).

Since

$$\alpha = P(Reject\ H_0|H_0\ is\ true) = P(\bar{x} < 0.7335\ when\ \mu = 0.75)$$

$$= P(\frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}} < \frac{0.7335 - 0.75}{\frac{0.06}{\sqrt{36}}}) = P(Z < -1.645) = 0.05$$

Next, we show how to compute the probability of type II error β, if $\mu = 0.70$ (as depicted in Figure 5.10).

$$\beta = P(Do\ Not\ Reject\ H_0|H_0\ is\ false) = P(Do\ Not\ Reject\ H_0|\mu = 0.70)$$

$$= P(\bar{x} \geq 0.7335|\mu = 0.7)$$

$$= P(\frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}} \geq \frac{0.7335 - 0.70}{\frac{0.06}{\sqrt{36}}}) = P(Z \geq 0.03355) = 1 - P(Z < 0.03355)$$

$$= 1 - 0.9996 = 0.0004$$

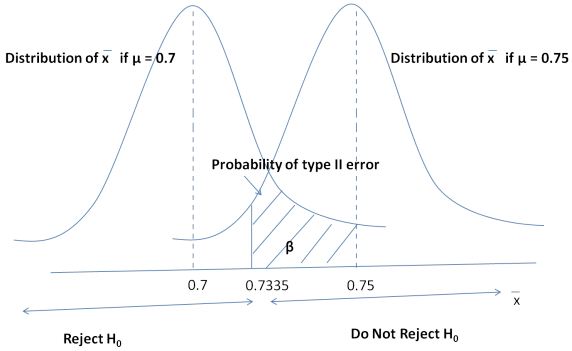

Figure 5.10: Probability of type II error.

Now, at $\alpha = 0.05$, the critical value Z_0 is -1.645. The test statistic

$$Z = \frac{(Sample\ value - Hypothesized\ value)}{Standard\ error} = \frac{\bar{x} - \mu_0}{\sigma_{\bar{x}}} = \frac{\bar{x} - \mu_0}{\frac{\sigma}{\sqrt{n}}}$$

and since $n > 30$, we can replace σ with the sample standard deviation s we have:

$$Z = \frac{0.745 - 0.75}{\frac{0.06}{\sqrt{36}}} = -0.05$$

Since the value of the test statistic (-0.05) is greater than the critical value $Z_0 = -1.645$, we fail to reject H_0. Indeed, there is not enough evidence to reject the claim that $\mu \geq \beta_h - \vartheta$. Therefore, we *correctly* classify value v as being (α_c,ϑ)-consistent in this case.

5.3 Empirical study

Suppose we are given a chance constraint $h : pr\{C\} \geq \beta_h$ over an infinite set of scenarios Ω_h, an assignment A in which $x = v$, a confidence level α_c, and a threshold error ϑ. The basic question that this thesis tries to answer is whether we could classify v as (α_c,ϑ)-consistent or not. The approach that we proposed in the previous section reformulates the problem as a statistical inference problem that could be solved using hypothesis testing. Let μ be the probability of constraint C restricted to assignment A. Since the number of scenarios Ω_h is infinite, such a quantity is unknown in general. In our approach, we state the Null hypothesis as $H_0 : \mu \geq \beta_h - \vartheta$ and the alternative as $H_1 : \mu < \beta_h - \vartheta$. Then, we generate a large enough sample S to guarantee a confidence level α_c. Next, we compute an appropriate test statistic (z-test statistic in our case) and use it to either accept or reject the Null hypothesis. If our Null hypothesis is accepted, we do classify v as (α_c,ϑ)-consistent, else as (α_c,ϑ)-inconsistent.

The main goal of this empirical study is to answer the following questions:

1. Can the hypothesis testing approach yield an acceptable correct classification rate in practice?

2. Is the theoretical sample size large enough to guarantee the given confidence level?

3. How does the hypothesis testing approach compare against the approach based on confidence intervals?

5.3.1 Experimental Settings

To be able to an exhaustive empirical study that can cover most situations in practice, we consider the following experimental set-up:

1. We choose an arbitrary value μ to be 0.5.

2. We choose the classical value 0.95 for α_c.

3. We consider the following sample sizes:

$$n \in \{50, 100, 200, \ldots, 1000, 1200, 1400, \ldots, 2000, 2500, \ldots,$$
$$5000, 6000, \ldots, 10000, 20000, 30000, 40000\}$$

   For each sample size we generate a sample S_n of 0's and 1's following the Bernoulli distribution whose $p = \mu$.

4. We vary β_h around the μ value as follows: $\beta_h \in \{\mu - 0.02, \mu - 0.01, \mu, \mu + 0.01, \mu + 0.02, \mu + 0.03, \mu + 0.04, \mu + 0.05, \mu + 0.06, \mu + 0.07, \mu + 0.08, \mu + 0.09\}$.

5. We vary ϑ between a very small value 0.001 and a larger one 0.01.

Note that the above set-up simulates in an exhaustive manner many practical situations. Indeed, we have the following:

- Each configuration of $\langle \mu, \beta_h, \alpha_c, \vartheta \rangle$ represents a practical situation of some chance constraint $h : pr\{C\} \geq \beta_h$ and some assignment A in which $x = v$. A value v is (α_c, ϑ)-consistent if $\mu \geq \beta_h - \vartheta$. Otherwise, v is (α_c, ϑ)-inconsistent.

- Each configuration $\langle S_n, \beta_h, \alpha_c, \vartheta \rangle$ represents the chance constraint h (i.e., configuration $\langle \mu, \beta_h, \alpha_c, \vartheta \rangle$) restricted to the subset of the scenarios in S_n. Our goal is to infer whether or not value v is (α_c, ϑ)-consistent or not using the hypothesis testing approach.

$n \backslash \beta_h = \mu +$	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
50	5.9%	8.5%	8.5%	8.3%	14.7%	14.7%	21.3%	21.3%	30.2%
100	6.9%	9.4%	13.2%	17.4%	22.4%	29.9%	37.1%	45.6%	54.8%
200	5.1%	9.8%	15.8%	23.2%	34.3%	45.9%	57.7%	67.3%	77%
300	10.1%	14%	22.7%	33.5%	47.8%	63.3%	75.8%	84.5%	91.1%
400	8.5%	16.9%	29%	43.2%	57.7%	72.8%	85.4%	92.5%	**97.6%**
500	11%	21.3%	35.9%	55.6%	71%	83.8%	92.7%	**97.6%**	**99.3%**
600	10.1%	21.7%	41.2%	58.2%	74.9%	87.5%	**95.4%**	**97.7%**	**99.9%**
700	10.2%	23.3%	41.6%	63.1%	81.1%	92.5%	**97.8%**	**99.1%**	**100%**
800	13.9%	29%	48.5%	70.3%	86.1%	**95.2%**	**98.2%**	**100%**	**100%**
900	11.6%	28.2%	50.8%	72.9%	89%	**96.9%**	**99.7%**	**100%**	**100%**
1000	14.1%	31.5%	55.1%	79.3%	92.4%	**98.1%**	**99.5%**	**100%**	**100%**
1200	15%	35.4%	61.4%	82.9%	**95%**	**98.6%**	**99.8%**	**100%**	**100%**
1400	14.7%	40.8%	67.2%	87.2%	**97%**	**99.6%**	**100%**	**100%**	**100%**
1600	16.5%	43%	73.1%	91.9%	**98.5%**	**99.8%**	**100%**	**100%**	**100%**
1800	16%	46.3%	78.5%	94.5%	**99.2%**	**99.9%**	**100%**	**100%**	**100%**
2000	20.5%	50%	81.5%	**96.6%**	**99.6%**	**100%**	**100%**	**100%**	**100%**
2500	20.8%	56.6%	86%	**97.6%**	**99.9%**	**100%**	**100%**	**100%**	**100%**
3000	24.1%	65%	94.7%	**99.3%**	**100%**	**100%**	**100%**	**100%**	**100%**
3500	27.1%	68.8%	**96%**	**99.8%**	**100%**	**100%**	**100%**	**100%**	**100%**
4000	29.6%	76.4%	**98.1%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
4500	33%	80.4%	**99%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
5000	33.5%	82.9%	**99%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
6000	41.4%	89.2%	**99.8%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
7000	42.4%	93.3%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
8000	49.6%	**95.7%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
9000	49%	**98.2%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
10000	51.3%	**98.5%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
20000	80.9%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
30000	91.1%	**98.5%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
40000	**97.3%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**

Figure 5.11: Correct classification rate of an inconsistent value for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.001$.

Finally, we generate randomly 1000 configurations $\langle S_n, \beta_h, \alpha_c, \vartheta \rangle$ by generating 1000 different S_n's. For each of the 1000 configurations (or experiments), we perform the hypothesis testing using our Null hypothesis H_0 and depending on whether we accept or reject H_0 we either classify v as (α_c, ϑ)-consistent or not.

5.3.2 Validation of the Hypothesis Testing Approach

Firstly, we will calculate the theoretical sample size needed to guarantee a confidence level of $\alpha_c = 0.95$ in terms of correct inference about a particular value v of being either (α_c,ϑ)-consistent or not. Recall that sample size , as described in section 5.2.2, is equal to:

$$n = \frac{(Z_\alpha + Z_\beta)^2 \sigma^2}{\delta^2}$$

Since, $\alpha_c = 0.95$, we set the significance level α to 0.05 which would restrict the

type I error, i,e, rejecting the Null hypothesis when it is true. Similarly, we set β to 0.05 to restrict the rate of type II errors which is failing to reject the Null hypothesis when it is false. Thus, the power of the test would be $1 - \beta = 0.95$.

The standard deviation σ is indeed unknown in our case. We should, therefore, estimate it by using a sample standard deviation s. We take a random sample of size 50 which has a binomial distribution and compute s as follows:

$$s = \sqrt{\frac{1}{50}\sum_{i=1}^{50}(x_i - \bar{x})^2}$$

where $\bar{x}$ is the sample mean. We computed the sample standard deviation of a random sample of size 50 and it happens to be 0.5036.

The value for Z_α is equal to Z_β which is -1.645. While the values for α and β in general control the accuracy of our test (i.e., accuracy is how close a measured value is to the actual true value.), the choice of δ seems to define how precise we would like our inference to be (i.e., precision is how close the measured values are to each other). Indeed, if we decrease δ, we do increase our precision. If we choose a very high precision and set δ to 0.001, the sample size needed in theory turns out to be a very large number:

$$n = \frac{(-1.645 + -1.645)^2 0.5036^2}{0.001^2} = 2\ 745\ 132$$

For a δ value of 0.01 instead, the theoretical sample size need is 27 451 whereas for a larger δ being 0.05, the sample size is just 1098.

Case I: $\langle \mu = 0.5, \alpha_c = 0.95, \vartheta = 0.001 \rangle$

In Figure 5.11, we report the CCR for the case where $\mu = 0.5$, $\alpha_c = 0.95$, $\vartheta = 0.001$, and $\beta_h \in \{\mu + 0.01, \ldots, \mu + 0.09\}$. The sample size has been varied as described in the Section 5.3. Each entry in Figure 5.11 represents a run of 1000 random experiments of hypothesis testing of our Null hypothesis where each experiment corresponds to a random configuration $\langle S_n, \beta_h, \alpha_c, \vartheta \rangle$. In all the cases in these experiments, the correct classification is to classify the values as (α_c, ϑ)-inconsistent. In fact, a misclassification in this case corresponds to a type II error since we accept a Null hypothesis that is false.

The following observations can be stated based on the results in Figure 5.11:

- When δ is set to 0.001, the theoretical sample size ($n = 2\ 745\ 132$) is too high and indeed guarantees that our inference method achieves a CCR of 95% or above under all parameter settings of μ, β_h and ϑ;

- When δ is set to 0.01, the theoretical sample size ($n = 27\ 451$) is almost large enough so that our inference method guarantees under all parameter settings of μ, β_h and ϑ to achieve above a 95% CCR with an exception when $\beta_h = \mu + 0.01$ which requires a sample size of 30000 or more;

- When δ is set to 0.05, the theoretical sample size ($n = 1098$) is too little to guarantee that our inference method achieves a CCR of 95% or above. Indeed, only when $\beta_h \geq \mu + 0.06$, we do achieve above a 95% CCR;

- As the distance between $\beta_h - \vartheta$ and μ gets closer, the number of samples needed to guarantee a 95% classification rate increases. As they get apart, the sample size needed to guarantee a 95% classification rate decreases. Indeed, the case that requires the largest sample size is when the distance between $\beta_h - \vartheta$ and μ is the smallest.

The results in Figure 5.11 show that type II errors are indeed sensitive to the value of δ since as δ decreases, so does the error rate. But if δ is chosen carefully, a 95% or above CCR when a value v is (α_c, ϑ)-inconsistent can be achieved.

In Figure 5.12, we report the CCR for the case where $\mu = 0.5$, $\alpha_c = 0.95$, $\vartheta = 0.001$, and $\beta_h \in \{\mu, \mu - 0.01, \mu - 0.02\}$. The sample size has been varied as described in the Secion 5.3. Each entry in Figure 5.12 represents a run of 1000 random experiments of hypothesis testing of our Null hypothesis where each experiments corresponds to a random configuration $\langle S_n, \beta_h, \alpha_c, \vartheta \rangle$. In all the cases in these experiments, the correct classification is to classify the values as (α_c, ϑ)-consistent. In fact, a misclassification in this case corresponds to a type I error since we reject a Null hypothesis that is true.

The results in Figure 5.12 show that we require a much smaller sample size than the theoretical ones in order to achieve a 95% CCR compared to the inconsistent case. Indeed, the results show that sample size of as large as 400 are enough to make the desired inference.

In Figure 5.13, we plot the number of samples that guarantee a CRR equal to or above 95% as a function of the distance between μ and β_h for $\vartheta = 0.001$. The message is indeed clear: as the distance between μ and $\beta_h - \vartheta$ gets closer and the values are (α_c, ϑ)-inconsistent, we require bigger sample sizes. Otherwise, with much less sample sizes, we do achieve the desired 95% or above CCR.

Case II: $\langle \mu = 0.5, \alpha_c = 0.95, \vartheta = 0.01 \rangle$

In Figure 5.14, we report the CCR for the case where $\mu = 0.5$, $\alpha_c = 0.95$, $\vartheta = 0.01$, and $\beta_h \in \{\mu + 0.01, \ldots, \mu + 0.09\}$. The only difference between the experiments in

$n \backslash \beta_h$	$\beta_h = \mu$	$\beta_h = \mu - 0.01$	$\beta_h = \mu - 0.02$
50	94.1%	**96.5%**	**96.5%**
100	94.8%	**96%**	**97.4%**
200	**96.9%**	**98.3%**	**99.2%**
300	94.9%	**97.5%**	**98.9%**
400	**96%**	**97.8%**	**99.3%**
500	**95.5%**	**98.4%**	**99.6%**
600	**95.2%**	**98.3%**	**99.5%**
700	**96%**	**98.9%**	**99.5%**
800	**95%**	**99.1%**	**99.8%**
900	**97%**	**99%**	**99.7%**
1000	**95.9%**	**99.4%**	**100%**
1200	**95.7%**	**99.4%**	**100%**
1400	**96.7%**	**99.7%**	**100%**
1600	**95.9%**	**99.2%**	**100%**
1800	**97.2%**	**100%**	**100%**
2000	**96.6%**	**99.9%**	**100%**
2500	**96.6%**	**99.7%**	**99.9%**
3000	**96.9%**	**99.9%**	**100%**
3500	**96.9%**	**99.8%**	**100%**
4000	**96.5%**	**100%**	**100%**
4500	**96%**	**99.8%**	**100%**
5000	**97.2%**	**100%**	**100%**
6000	**97%**	**100%**	**100%**
7000	**95.8%**	**100%**	**100%**
8000	**97.5%**	**100%**	**100%**
9000	**97.8%**	**100%**	**100%**
10000	**97.8%**	**100%**	**100%**
20000	**96.9%**	**100%**	**100%**
30000	**97.8%**	**100%**	**100%**
40000	**98%**	**100%**	**100%**

Figure 5.12: Correct classification rate of a consistent value for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.001$.

Figure 5.14 and Figure 5.11 is just a higher ϑ from 0.001 to 0.01 instead. Note that, the case where $\beta_h = \mu + 0.01$ is no longer covered in this case as the values become (α_c,

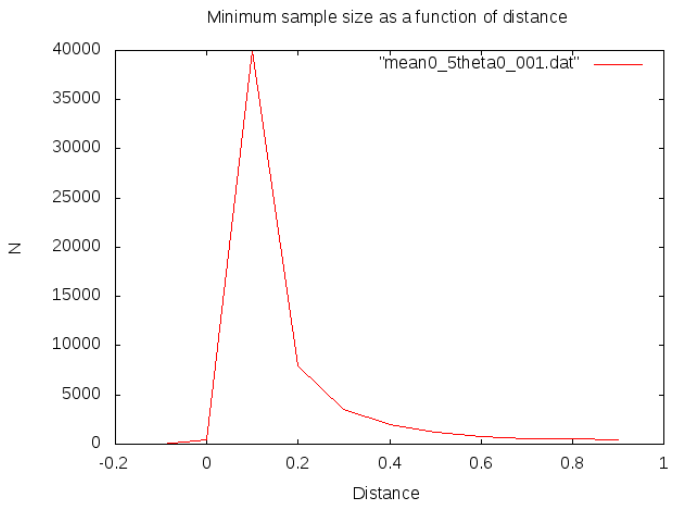

Figure 5.13: The x-axis is the distance $\beta_h - \mu$ whereas the y-axis is the number of sample above which we always achieve a correct classification rate greater than 95%. This plot is for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.001$.

ϑ)-consistent. We observe that the results are very similar to the ones in Figure 5.11 except that when δ is set to 0.01, we do achieve a 95% or above CCR in all cases.

In Figure 5.15, we report the CCR for the case where $\mu = 0.5$, $\alpha_c = 0.95$, $\vartheta = 0.01$, and $\beta_h \in \{\mu - 0.01, \mu, \mu + 0.01.\mu + 0.02\}$. The only difference between the experiments in Figure 5.12 and the ones in Figure 5.15 is the change in ϑ to 0.01 instead of 0.001.

We observe that for the case $\beta_h = \mu + 0.01$ we require a sample size close to the theoretical one for $\delta = 0.01$ in order to guarantee a CCR of 95% or above. Otherwise, we relatively small size is needed to achieve the desired CCR ($n = 50$). Again, we notice that the hardest case is when μ is closest to $\beta_h - \vartheta$.

Figure 5.16 is the same as Figure 5.13 for $\vartheta = 0.01$ instead. We clearly see the same pattern except that now the largest sample sizes are needed when μ approaches $\beta_h - \vartheta$ and the values are (α_c,ϑ)-consistent. Note that, our method based on hypothesis testing is quite robust against variations in the values of ϑ.

5.3.3 Comparing the Hypothesis Testing Approach Against The Confidence Intervals Approach

The approaches in Chapter 4 and the one in this chapter address the research problem in quite different ways. But, overall, the approach based on hypothesis testing is a superior and more robust method for the following reasons:

- First of all, in the confidence interval approaches in Chapter 4, there exists always

$n\backslash\beta_h = \mu+$	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
50	5.9%	8.5%	8.5%	14.7%	14.7%	14.7%	21.3%	21.3%
100	6.9%	9.4%	13.2%	17.4%	22.4%	29.9%	37.1%	45.6%
200	6.8%	9.8%	15.8%	23.2%	34.3%	45.9%	57.7%	67.3%
300	10.1%	15.6%	25.8%	38.9%	52.5%	67.4%	78.7%	86.1%
400	10.4%	12.5%	29%	43.2%	57.7%	72.8%	87.9%	94%
500	12.6%	21.3%	35.9%	55.6%	71%	86.1%	94.1%	**98%**
600	11.9%	24.3%	43.6%	61.1%	77.2%	89%	**95.6%**	**99.2%**
700	11.8%	24.9%	44.2%	65.4%	83.2%	93.1%	**98.1%**	**100%**
800	13.9%	29%	48.5%	70.3%	86.1%	**95.2%**	**98.3%**	**100%**
900	13.8%	31.2%	53.1%	75.1%	90.9%	**97.5%**	**99.7%**	**100%**
1000	15.6%	33.5%	57%	80.4%	93.4%	**98.2%**	**99.5%**	**100%**
1200	17.1%	33.3%	63.6%	89.3%	**95.4%**	**99%**	**99.8%**	**100%**
1400	19.2%	44.2%	71.1%	93.6%	**97.8%**	**99.6%**	**100%**	**100%**
1600	16.9%	46.8%	76.7%	94.9%	**98.5%**	**99.9%**	**100%**	**100%**
1800	22.3%	48%	79.3%	**97.1%**	**99.4%**	**100%**	**100%**	**100%**
2000	22.9%	53.3%	84.2%	**99.5%**	**99.9%**	**100%**	**100%**	**100%**
2500	26.6%	59.6%	87.6%	**99.8%**	**100%**	**100%**	**100%**	**100%**
3000	31.1%	69.4%	**95.7%**	**100%**	**100%**	**100%**	**100%**	**100%**
3500	34%	72.5%	**97.1%**	**100%**	**100%**	**100%**	**100%**	**100%**
4000	37.8%	80.7%	**98.6%**	**100%**	**100%**	**100%**	**100%**	**100%**
4500	37.8%	85.4%	**99.2%**	**100%**	**100%**	**100%**	**100%**	**100%**
5000	37.8%	86.4%	**99.6%**	**100%**	**100%**	**100%**	**100%**	**100%**
6000	47.2%	92%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
7000	48.2%	**95.2%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
8000	57.7%	**96.9%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
9000	56.3%	**98.7%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
10000	60%	**99.1%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
20000	87.7%	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
30000	**95.7%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
40000	**98.9%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**

Figure 5.14: Correct classification rate of an inconsistent value for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.01$.

either in the first method or in the second one a case such that it is impossible to properly classify a value v as being (α_c,ϑ)-consistent or not with probability α_c or above in theory and in practice. However, in the approach based on composite hypothesis testing, we can always classify value v as being (α_c,ϑ)-consistent or not with probability α_c or above in theory (and in practice when the sample size is carefully chosen).

- Secondly, in the hypothesis testing approach, we require a much smaller sample size in order to achieve a CCR of at least 95% when compared to either the first or the second method of Chapter 4. In most case, a sample size of as small as 400 is enough for the hypothesis testing approach to achieve the desired CCR when μ is not so close to $\beta_h - \vartheta$. When, μ approaches $\beta_h - \vartheta$, whereas the approaches based on confidence intervals may require sample sizes in the order of millions,

$n \backslash \beta_h$	$\beta_h = \mu + 0.01$	$\beta_h = \mu$	$\beta_h = \mu - 0.01$	$\beta_h = \mu - 0.02$
50	94.1%	**96.5%**	**96.5%**	**98.5%**
100	94.8%	**96%**	**97.4%**	**98.5%**
200	**96.1%**	**97.9%**	**98.9%**	**99.2%**
300	94.9%	**97.5%**	**98.9%**	**99.6%**
400	**95.3%**	**97.7%**	**99.6%**	**99.7%**
500	94.7%	**98.7%**	**99.4%**	**99.9%**
600	**95.6%**	**98.8%**	**99.8%**	**99.9%**
700	**95%**	**98.9%**	**99.6%**	**100%**
800	**96.6%**	**99.1%**	**100%**	**100%**
900	94.7%	**99.3%**	**99.9%**	**100%**
1000	**95.3%**	**99.7%**	**100%**	**100%**
1200	**96.2%**	**99%**	**100%**	**100%**
1400	94.5%	**99.8%**	**100%**	**100%**
1600	**96%**	**99.8%**	**99.9%**	**100%**
1800	**95.8%**	**99.6%**	**100%**	**100%**
2000	**95.7%**	**99.6%**	**100%**	**100%**
2500	**95.8%**	**100%**	**100%**	**100%**
3000	94.%	**99.9%**	**100%**	**100%**
3500	**96.2%**	**100%**	**100%**	**100%**
4000	**95.9%**	**100%**	**100%**	**100%**
4500	**95%**	**99.9%**	**100%**	**100%**
5000	**96.2%**	**100%**	**100%**	**100%**
6000	**95.9%**	**100%**	**100%**	**100%**
7000	**95%**	**100%**	**100%**	**100%**
8000	**96.2%**	**100%**	**100%**	**100%**
9000	**96.9%**	**100%**	**100%**	**100%**
10000	**96.7%**	**100%**	**100%**	**100%**
20000	94.3	**100%**	**100%**	**100%**
30000	**96%**	**100%**	**100%**	**100%**
40000	**96%**	**100%**	**100%**	**100%**

Figure 5.15: Correct classification rate of a consistent value for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.01$.

the approach based on hypothesis testing does not require more than 40000.

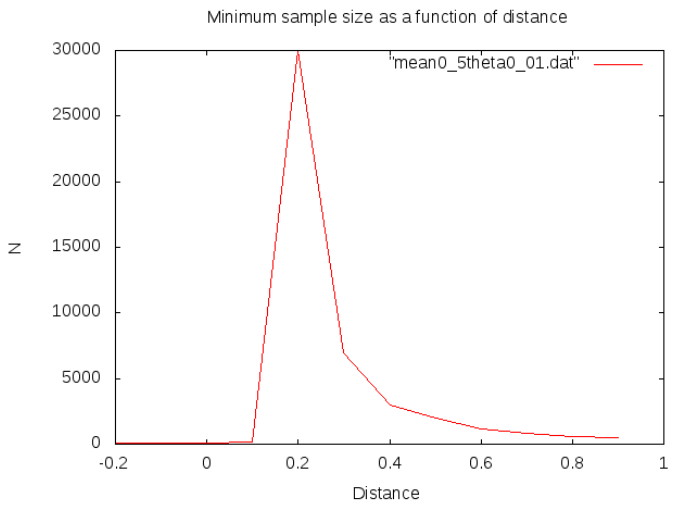

Figure 5.16: The x-axis is the distance $\beta_h - \mu$ whereas the y-axis is the number of sample above which we always achieve a correct classification rate greater than 95%. This plot is for $\mu = 0.5$, $\alpha_c = 0.95$, and $\vartheta = 0.01$.

5.4 Conclusion

In this chapter by assuming a large enough sample size, the satisfaction probability that we are trying to estimate from an infinite policy tree can be assumed to follow a Normal distribution rather than the Bernoulli one. Because of this normality assumption, we were able to formulate the problem of classifying whether or not a value v is (α_c,ϑ)-consistent by means of composite hypothesis testing where the composite Null hypothesis H_0 is stated as $\mu \geq \beta_h - \vartheta$ and the alternative composite hypothesis H_1 is stated as $\mu < \beta_h - \vartheta$. We have shown that an appropriate large enough smaple size that guarantees a CCR of at least 95% can be computed by Equation 5.1 in which: (1) the value of α is set to $1 - \alpha_c$ to limit the rate of type I errors, i.e., the rate of misclassifying a truly (α_c,ϑ)-consistent value as (α_c,ϑ)-inconsistent; (2) the value of β is set to $1 - \alpha_c$ to also limit the rate of type II error, i.e., the rate of misclassifying a truly (α_c,ϑ)-inconsistent value as (α_c,ϑ)-consistent; and (3) the value for δ was tuned in the experiments in order to reveal the best value in practice. The extensive empirical study have shown that the proposed method based on composite hypothesis testing is superior to the previous methods based on confidence intervals in terms of: (1) for a carefully chosen δ, we are able to do inference that achieves at least 95% CCR in all cases; (2) we do not have any regions in which we are unable to guarantee at least a 95% CCR; and (3) in most cases, a moderate sample size is enough and, as shown in Figure 5.13 and Figure 5.16, the largest sample size needed –when μ is very close to

$\beta_h - \vartheta$ — is no larger than 40000. In conclusion, the proposed method in this chapter based on hypothesis testing is an effective, robust, and practical method that is able to achieve in all cases a CCR of at least 95%.

Chapter 6

Conclusions and Future Work

In this chapter, we provide a detailed summary of the contributions of this thesis in Section 6.1. Then, in Section 6.2, we discuss the results and emphasise on the shortcomings of the thesis.

6.1 Summary

In this thesis, we take a different approach towards computing an (α_c,ϑ)-solution to a single-stage SCSP P. Instead of reformulating the original infinite SCSP P into a finite one restricted to a subset of scenarios $\hat{P}$ and use the standard search techniques to solve the resulting SCSP $\hat{P}$, we propose to directly work with the infinite SCSP P by lifting the inference that guides the search in the deterministic case to an inference that reasons about the uncertainties in the presence of an infinite set of scenarios. To this end, we focus on each chance constraint of an infinite SCSP on its own and propose a novel consistency property that lifts the notion from the deterministic case to the stochastic one. Such a notion of consistency is named as statistical consistency and in particular we propose (α_c,ϑ)-consistency as a specific instance of such consistency. The concept of (α_c,ϑ)-consistency is tightly related to the concept of (α_c,ϑ)-solutions proposed in [RHTP15] but fundamentally differs from it in the sense that while (α_c,ϑ)-solutions consider the whole SCSP as a whole, (α_c,ϑ)-consistency is focused on each chance constraint separately. Secondly, while (α_c,ϑ)-solutions are found through reformulation by using the standard search and inference techniques as advocated in [HRTP12], (α_c,ϑ)-consistency is a building block that be used to design new families of search algorithms that reasons about the infinite SCSP without any reformulation and directly solves it despite the infinite number of scenarios. Finally, while finding (α_c,ϑ)-solutions is sensitive to a large sample size, enforcing (α_c,ϑ)-consistency is not.

Suppose we are given an infinite chance constraint $h : pr\{C\} \geq \beta_h$ over an infinite

set of scenarios Ω_h, an assignment A in which $x = v$, a confidence level α_c, and a threshold error ϑ. The basic question that this thesis tries to answer is whether we could classify v as (α_c,ϑ)-consistent or not. The pillars upon which (α_c,ϑ)-consistency is based are three: (1) Since the number of scenarios in Ω_h is infinite, one cannot see a general method that can work for any chance constraint to determine whether or not a particular value v is consistent with respect to Ω_h. Thus, the first pillar of the notion of (α_c,ϑ)-consistency advocates using only a finite subset of scenarios in order to make a statistical inference about value v being consistent with respect to the infinite set Ω_h of scenarios; (2) Since, we are considering a subset of scenarios, then our inference will necessarily be in error. The second pillar of the notion of (α_c,ϑ)-consistency is the confidence probability α_c we wish to associate with how confident we are in our statistical inference; and (3) the last pillar is the notion of the error threshold value ϑ that specifies the margin of errors we are ready to tolerate while restricting our statistical inference to only a subset of scenarios.

To be able to enforce (α_c,ϑ)-consistency, this thesis proposed three main methods that can be employed after a Bonferroni correction to the confidence probability α_c: two methods are based on confidence intervals whereas the third one is based on composite hypothesis testing. The first approach we propose and experimentally validate is inspired by the way (α_c,ϑ)-solution were sought through sampling in [RHTP15]. As a statistical inference method, it is shown in this thesis, both theoretically and experimentally, that a CCR of at least $\alpha_c 100\%$ can be achieved as long as $\mu \notin]\beta_h - \vartheta, \beta_h + \vartheta[$. Otherwise, the proposed method does not have any guarantees to achieve a desired CCR. Furthermore, in the case when $\mu = \beta_h - \vartheta$, the first approach is flawed and always makes the wrong inference. Next, based on an analysis of the first approach by taking a different perspective which focuses on building a confidence interval around the sample mean $\bar{x}$ rather than around β_h as in the first approach, we propose our second method. In the second method, as in the first one, there is still a critical case in which we cannot guarantees to achieve the desired CCR. Yet, we suggested two versions of the second method to partially remedy this issue: (1) In the first version we tolerate the errors of misclassifying an inconsistent value as being consistent in the critical case (i.e., we are ready to endure a higher rate of type I errors); (2) In the second version we tolerate the errors of misclassifying a consistent value as being inconsistent in the critical case (i.e., we are ready to endure a higher rate of type II errors). The second method is shown theoretically and experimentally to be better than the first one since it makes more correct inference at a CCR of at least $\alpha_c 100\%$ and also overcomes the flaw of the first method when $\mu = \beta_h - \vartheta$. Finally, our last method is a novel approach based on composite hypothesis testing. In this approach, we assume a large enough sample

size that would allow us to consider the distribution of the satisfaction probability we are trying to estimate as being Normal instead of Bernoulli. Therefore, we were able to formulate our problem of classifying whether or not a value v is (α_c,ϑ)-consistent through means of composite hypothesis testing where the Null composite hypothesis is stated as $H_0 : \mu \geq \beta_h - \vartheta$ and the alternative composite one as $H_1 : \mu < \beta_h - \vartheta$. We have shown that an appropriate large enough sample size that guarantees a CCR of at least $\alpha_c 100\%$ can computed by using Equation 5.1. The extensive empirical study has shown that the proposed method based on composite hypothesis testing is superior to the previous ones based on confidence interval since, for a carefully chosen δ, we can do inference that achieves a CCR of at least $\alpha_c 100\%$ in all cases. We do not have any case in which we are unable to guarantee at least $\alpha_c 100\%$ CCR. Furthermore, a moderate sample size is enough in most situations to guarantee an $\alpha_c 100\%$ CCR. For all these reasons, we can claim that the method based on composite hypothesis testing is an effective, robust, and practical method.

In Summary, this thesis seems to have answered the two main research questions stated in the introduction in a satisfactory manner. The first question:

How to extend the notion of consistency to an *infinite chance* constraint?

has been answered by offering the notion of (α_c,ϑ)-consistency:

Definition 13. *Given a chance constraint h. A value v in the domain of $x \in X_h$ is (α_c, ϑ)-GAC iff there exists an assignment A in which $x = v$ and, with confidence level α_c,*

$$\sum_{s \in \Omega_h} B_h^s pr(s) \geq \beta_h - \vartheta$$

The second question:

How to enforce statistical consistency?

has been answered by offering three methods with varying degrees of effectiveness as depicted in Figure 6.1.

6.2 Discussion and Future Work

Some of the shortcomings and possible future works of this thesis can be summarised as follows:

- The two methods based on confidence intervals fail to fully achieve the goal of guaranteeing a CCR of at least $\alpha_c 100\%$ CCR in general. The main reason is due

Method	Strength	Weakness
Approach 1 in Section 4.2	Achieves a desired CCR when $\mu \notin]\beta_h - \vartheta, \beta_h + \vartheta[$	cannot guarantee the desired CCR when $\mu \in [\beta_h - \vartheta, \beta_h + \vartheta[$.
Approach 2: version I in Section 4.3	Achieves a desired CCR when $\beta_h \notin]\bar{x} + \frac{\vartheta}{2}, \bar{x} + \frac{3\vartheta}{2}]$ or $\beta_h \in]\bar{x} + \frac{\vartheta}{2}, \bar{x} + \frac{3\vartheta}{2}]$ and $\mu \geq \beta_h - \vartheta$	cannot guarantee the desired CCR when $\beta_h \in]\bar{x} + \frac{\vartheta}{2}, \bar{x} + \frac{3\vartheta}{2}]$ and $\mu < \beta_h - \vartheta$.
Approach 2: version II in Section 4.3	Achieves a desired CCR when $\beta_h \notin]\bar{x} + \frac{\vartheta}{2}, \bar{x} + \frac{3\vartheta}{2}]$ or $\beta_h \in]\bar{x} + \frac{\vartheta}{2}, \bar{x} + \frac{3\vartheta}{2}]$ and $\mu < \beta_h - \vartheta$	cannot guarantee the desired CCR when $\beta_h \in]\bar{x} + \frac{\vartheta}{2}, \bar{x} + \frac{3\vartheta}{2}]$ and $\mu \geq \beta_h - \vartheta$.
Approach 3 in Section 5.2	Achieves a desired CCR when the sample size is carefully chosen	Sensitive to the choice of δ.

Figure 6.1: Summary of the three proposed methods

to the fact that if $\beta_h - \vartheta$ falls within the confidence interval, then based on the evidence we are unable to figure out whether or not μ is to the right of $\beta_h - \vartheta$ or to the left of it. It seems that no matter how small we make the width of such a CI, we always have such kind of situations as stated in Theorem 3.

- Another drawback is that when we want to enforce (α_c,ϑ)-consistency, the confidence probability needs to be adjusted using the Bonferroni correction. This may lead to very high adjusted values which in turn will result in very high sample sizes. So, how to deal with this bottelneck in practice? are there any less conservative approaches that one can try instead of the Bonferroni correction?

- The last method based on composite hypothesis testing is quite sensitive to the choice of δ. So, a possible future research direction is to tackle this issue. Indeed, is there a relationship between δ and ϑ that one can exploit?

Bibliography

[AC98] A. Agresti and B. A. Coull. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*, 52(2):119–126, 1998. 46

[Apt03] Krzysztof Apt. *Principles of Constraint Programming*. Cambridge University Press, New York, NY, USA, 2003. 29, 30

[AS08] S. Ahmed and A. Shapiro. Solving chance-constrained stochastic programs via sampling and integer programming. In Z.-L. Chen and S. Raghavan, editors, *Tutorials in Operations Research*, chapter 12, pages 261–269. INFORMS, 2008. 22

[Ask81] R. G. Askin. A procedure for production lot sizing with probabilistic dynamic demand. *AIIE Transactions*, 13:132–137, 1981. 9

[ASS02] Shabbir Ahmed, Alexander Shapiro, and Er Shapiro. The sample average approximation method for stochastic programs with integer recourse. *SIAM Journal of Optimization*, 12:479–502, 2002. 21

[AW70] M. Avriel and A. C. Williams. The value of information and stochastic programming. *Operations Research*, 18(5):947–954, 1970. 9

[BCDP07] Nicolas Beldiceanu, Mats Carlsson, Sophie Demassey, and Thierry Petit. Global constraint catalogue: Past, present and future. *Constraints*, 12(1):21–62, 2007. 32

[BCNN00] Jerry Banks, John S. Carson, Barry L. Nelson, and David M. Nicol. *Discrete-Event System Simulation (3rd Edition)*. Prentice Hall, 3 edition, 2000. 62, 72

[BFR99] Christian Bessière, Eugene C. Freuder, and Jean-Charles Régin. Using constraint metaknowledge to reduce arc consistency computation. *Artif. Intell.*, 107(1):125–148, 1999. 32

[BH03] Christian Bessière and Pascal Van Hentenryck. To be or not to be ... a global constraint. In *Principles and Practice of Constraint Programming - CP 2003, 9th International Conference, CP 2003, Kinsale, Ireland, September 29 - October 3, 2003, Proceedings*, pages 789–794, 2003. 32

[BHHW07] C. Bessiere, E. Hebrard, B. Hnich, and T. Walsh. The complexity of reasoning with global constraints. *Constraints*, 12(2):239–259, 2007. 34

[BL97] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Verlag, New York, 1997. 9, 10

[BM06] K. N. Brown and I. Miguel. Uncertainty and change. In F. Rossi, P. van Beek, and T. Walsh, editors, *Handbook of Constraint Programming*, chapter 21. Elsevier, 2006. 19

[BPS99] S. C. Brailsford, C. N. Potts, and B. M. Smith. Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119:557–581, 1999. 26

[BR97] Christian Bessière and Jean-Charles Régin. Arc consistency for general constraint networks: Preliminary results. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97, Nagoya, Japan, August 23-29, 1997, 2 Volumes*, pages 398–404, 1997. 31

[BR98] Christian BessiËre and Jean-Charles RÈgin. Local consistency on conjunctions of constraints. In *ECAI-98 Workshop on Non-binary Constraints*, 1998. 32

[Bra12a] Martin Branda. Sample approximation technique for mixed-integer stochastic programming problems with several chance constraints. *Operations Research Letters*, 40(3):207–211, May 2012. 21, 22

[Bra12b] Martin Branda. Stochastic programming problems with generalized integrated chance constraints. *Optimization*, 61(8):949–968, July 2012. 21

[Bra13] Martin Branda. On relations between chance constrained and penalty function problems under discrete distributions. *Mathematical Methods of Operations Research*, 77(2):265–277, 2013. 22

[BS06] T. Balafoutis and K. Stergiou. Algorithms for stochastic csps. In *Principles and Practice of Constraint Programming, CP 2006, Proceedings*, pages 44–58, 2006. 19

[BT88] J. H. Bookbinder and J. Y. Tan. Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science*, 34:1096–1108, 1988. 9

[CC59] A. Charnes and W. W. Cooper. Chance-constrainted programming. *Management Science*, 6(1):73–79, 1959. 9

[CC63] A. Charnes and W. W. Cooper. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research*, 11(1):18–39, 1963. 9

[Che52] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4), 1952. 22

[CP34] C. Clopper and E. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. 38, 40, 46

[DB97] Romuald Debruyne and Christian Bessiere. Some practicable filtering techniques for the constraint satisfaction problem. In *In Proceedings of IJCAI97*, pages 412–417, 1997. 30

[FLMCS95] H. Fargier, J. Lang, R. Martin-Clouaire, and T. Schiex. A constraint satisfaction framework for decision under uncertainty. In Philippe Besnard and Steve Hanks, editors, *UAI '95: Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, August 18-20, 1995, Montreal, Quebec, Canada*, pages 167–174. Morgan Kaufmann, 1995. 19

[GB97] S.B. Green and M.A. Babyak. Control of type i errors with multiple tests of constraints in structural equation modeling. *Multivariate Behavioral Research*, 32(1):39–51, 1997. 59

[Gra99] S. C. Graves. A Single-Item Inventory Model for a Non-Stationary Demand Process. *Manufacturing & Service Operations Management*, 1:50–61, 1999. 9

[GS77] S. K. Gupta and J. K. Sengupta. Decision rules in production planning under chance-constrained sales. *Decision Sciences*, 8:521–533, 1977. 9

[GW74] S. J. Gartska and R. J. B. Wets. On decision rules in stochastic programming. *Mathematical Programming*, 7:117–143, 1974. 9

[HC88] P. Van Hentenryck and J. P. Carillon. Generality vs. specificity: an experience with ai and or techniques. In *Proceedings of the 7th National Conference on Artificial Intelligence. (AAAI-88) St. Paul, MN, August 21-26*, pages 660–664, 1988. 20

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. 22

[HRTP10] B. Hnich, R. Rossi, S. A. Tarim, and S. Prestwich. A survey on cp-ai-or hybrids for decision making under uncertainty. In M. Milano and P. Van Hentenryck, editors, *Hybrid Optimization: the 10 years of CP-AI-OR*, Lecture Notes in Computer Science. Springer, 2010. 21

[HRTP11] B. Hnich, R. Rossi, S. A. Tarim, and S. Prestwich. A survey on CP-AI-OR hybrids for decision making under uncertainty. In Pascal van Hentenryck and Michela Milano, editors, *Hybrid Optimization*, volume 45 of *Springer Optimization and Its Applications*, chapter 7, pages 227–270. Springer New York, New York, NY, 2011. 19

[HRTP12] Brahim Hnich, Roberto Rossi, S. Armagan Tarim, and Steven David Prestwich. Filtering algorithms for global chance constraints. *Artif. Intell.*, 189:69–94, 2012. 1, 6, 19, 20, 21, 33, 34, 40, 86

[HSD98] Pascal Van Hentenryck, Vijay Saraswat, and Yves Deville. Design, implementation, and evaluation of the constraint language cc(fd). *The Journal of Logic Programming*, 37(13):139 – 164, 1998. 18

[ILO07] Inc. ILOG. *OPL Studio 3.7 Users Manual*. ILOG, Inc., Incline Village, NV, 2007. 26

[Knu68] Donald E. Knuth. *The Art of Computer Programming, Volume I: Fundamental Algorithms*. Addison-Wesley, 1968. 28

[KSHDM01] A. J. Kleywegt, A. Shapiro, and T. Homem-De-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal of Optimization*, 12(2):479–502, 2001. 21

[Kuh05] D. Kuhn. *Generalized bounds for convex multistage stochastic programs*, volume 584 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, 2005. 9

[KW94] P. Kall and S. W. Wallace. *Stochastic Programming*. John Wiley & Sons, 1994. 9

[LA08] James Luedtke and Shabbir Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM J. on Optimization*, 19(2):674–699, July 2008. 22

[LP01] I. J. Lustig and J. F. Puget. Program does not equal program: Constraint programming and its relationship to mathematical programming. *Interfaces*, 31:29–53, 2001. 20

[LS99] Tung Liu and Courtenay C. Stone. A critique of one-tailed hypothesis test procedures in business and economics statistics textbooks. *The Journal of Economic Education*, 30(1):59–63, 1999. 69

[LSW06] Jeff Linderoth, Alexander Shapiro, and Stephen Wright. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research*, 142(1):215–241, February 2006. 21

[LtOpt94] F. Laburthe and the OCRE project team. Choco: Implementing a cp kernel. Technical report, Bouygues e-Lab, France, 1994. 26

[MM88] Roger Mohr and Gérald Masini. Good old discrete relaxation. In Yves Kodratoff, editor, *8th European Conference on Artificial Intelligence*, pages 651–656, Munich, Germany, 1988. Pitmann Publishing. 17

[MR03] D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley and Sons, 2003. 62, 72, 73, 98

[NW88] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley and Sons, 1988. 20

[Pap84] A. Papoulis. *Probability, Random Variables, and Stochastic Processes (2nd ed.)*. McGraw-Hill, New York, 1984. 41, 108

[PAS09] B. K. Pagnoncelli, S. Ahmed, and A. Shapiro. Sample average approximation method for chance constrained programming: Theory and applications. *Journal of Optimization Theory and Applications*, 142(2):399–416, August 2009. 22

[Por02] E. L. Porteus. *Foundations of Stochastic Inventory Theory*. Stanford University Press, Stanford, CA, 2002. 9

[PS08] I. N. Pujawan and E. A. Silver. Augmenting the lot sizing order quantity when demand is probabilistic. *European Journal of Operational Research*, 127(3):705–722, August 2008. 9

[PTRH09] S. D. Prestwich, S. A. Tarim, R. Rossi, and B. Hnich. Evolving parameterised policies for stochastic constraint programming. In Ian P. Gent, editor, *CP 2009, Proceedings*, volume 5732 of *Lecture Notes in Computer Science*, pages 684–691. Springer, 2009. 21

[R C13] R Core Team. *R: A language amd Environment for Statistical Computing*. R Foundation for Statistical Computing, 2013. 108

[Reg94] J.-C. Regin. A filtering algorithm for constraints of difference in csps. In *Proceedings of the 12th National Conference on Artificial Intelligence, Volume 1. Seattle, WA, USA, July 31 - August 4*, pages 362–367. AAAI Press, 1994. 33

[Rég96] Jean-Charles Régin. Generalized arc consistency for global cardinality constraint. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, AAAI'96, pages 209–215. AAAI Press, 1996. 33

[RHTP11] Roberto Rossi, Brahim Hnich, S. Armagan Tarim, and Steven David Prestwich. Finding (α,ϑ)-solutions via sampled scsps. In *IJCAI*, pages 2172–2177, 2011. 5, 21, 39, 40

[RHTP15] Roberto Rossi, Brahim Hnich, S. Armagan Tarim, and Steven Prestwich. Confidence-based reasoning in stochastic constraint programming. *Artificial Intelligence*, 228:129 – 152, 2015. viii, ix, 5, 21, 35, 36, 37, 39, 40, 41, 42, 43, 46, 47, 48, 52, 53, 86, 87

[RvBW06] F. Rossi, P. van Beek, and T. Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York, NY, USA, 2006. 16

[Sam65] S. M. Samuels. On the number of successes in independent trials. *Annals of Mathematical Statistics*, 36:1272–1278, 1965. 108

[Sca60] H. Scarf. *The optimality of (s,S) policies in dynamic inventory problem*. in Arrow, Karlin and Suppes, Mathematical Methods in Social Sciences 1959, Stanford Unversity Press, 1960. 9

[Sen72] J. K. Sengupta. *Stochastic Programming: Methods and Applications*. North-Holland, Amsterdam, 1972. 9

[Sil78] E. A. Silver. Inventory control under a probabilistic time-varying demand pattern. *AIIE Transactions*, 10:371–379, 1978. 9

[SWdK04] S. R. Smits, M. Wagner, and T. G de Kok. Determination of an order-up-to policy in the stochastic economic lot scheduling model. *International Journal of Production Economics*, 90:377–389, 2004. 9

[Tem07] H. Tempelmeier. On the stochastic uncapacitated dynamic single-item lotsizing problem with service level constraints. *European Journal of Operational Research*, 127(1):184–194, August 2007. 9

[THPR08] S. A. Tarim, B. Hnich, S. D. Prestwich, and R. Rossi. Finding reliable solution: Event-driven probabilistic constraint programming. *Annals of Operations Research*, 2008. 19

[TK06] S. A. Tarim and B. G. Kingsman. Modelling and Computing (R^n,S^n) Policies for Inventory Systems with Non-Stationary Stochastic Demand. *European Journal of Operational Research*, 174:581–599, 2006. 9

[TMW06] S. A. Tarim, S. Manandhar, and T. Walsh. Stochastic constraint programming: A scenario-based approach. *Constraints*, 11(1):53–80, 2006. 1, 2, 19, 21, 40

[TS08] S. A. Tarim and B. Smith. Constraint Programming for Computing Non-Stationary (R,S) Inventory Policies. *European Journal of Operational Research*, 189:1004–1021, 2008. 9

[Tsa93] E. Tsang. *Foundations of Constraint Satisfaction*. Computation in cognitive science. Academic Press, London, 1993. 16, 17, 23

[Vaj72] S. Vajda. *Probabilistic Programming*. Academic Press, New York, 1972. 9

[vB06] Peter van Beek. Backtracking search algorithms, 2006. 31

[Vol93] Stein Emil Vollset. Confidence intervals for a binomial proportion. *Statistics in Medicine*, 12(9):809–824, 1993. 46

[Wal01] Toby Walsh. Permutation problems and channelling constraints. In *Logic for Programming, Artificial Intelligence, and Reasoning, 8th International Conference, LPAR 2001, Havana, Cuba, December 3-7, 2001, Proceedings*, pages 377–391, 2001. 30

[Wal02] T. Walsh. Stochastic constraint programming. In *European Conference on Artificial Intelligence, ECAI'2002, Proceedings*, pages 111–115, 2002. 1, 2, 19

[Wol98] L. A. Wolsey. *Integer Programming*. John Wiley, New York, 1998. 11

Appendix A

Formal Background

In this appendix we outline all the necessary formal background needed to be able to understand the technical details of this thesis. In Section A.1, we review the basic definition and concepts in probability theory and statistics. Next, in Section A.2, we describe various sampling techniques. Note that most of the material in this Chapter is a based on [MR03].

A.1 Probability Theory and Statistics

Statistics is concerned with making inference about populations and population characteristics [MR03]. Experiments are conducted with results that are subject to chance. The tossing of a coin is an example of a *statistical experiment*, a term that is used to describe any process that generates a set of data. A set of *observations* is obtained by repeating the experiment several times. The set of all possible outcomes of a statistical experiment is called the *sample space* and is represented by the symbol S. Each outcome in a sample space is called an *element* or a *member* of the sample space. An *event* is a subset of a sample space or simply a sample point. The *complement* of an event A with respect to S is the subset of all events of S that are not in A. Two events A and B are *mutually exclusive* or *disjoint* if $A \cap B = \emptyset$.

The *probability of an event* A is the sum of the weights of all sample points in A. Therefore,

$$0 \leq P(A) \leq 1, \quad P(\emptyset) = 0, \quad P(S) = 1$$

Furthermore, if $A_1, A_2, \ldots$ is a sequence of mutually exclusive events, then

$$P(A_1 \cup A_2 \cup \ldots) = P(A_1) + P(A_2) + \ldots$$

If an experiment can result in any one of N different equally likely outcomes, and if

exactly n of these outcomes correspond to event A, then the probability of event A is

$$P(A) = \frac{n}{N}$$

If A and B are any two events, then

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

For the events A, B, and C,

$$P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C)$$

The probability of an event B occurring when it is known that some event A has occurred is called a *conditional probability* and is denoted by $P(B|A)$. We have:

$$P(B|A) = \frac{P(A \cap B)}{P(A)}, \ if \ \ P(A) > 0$$

Two events A and B are *independent* if and only if: $P(B|A) = P(B)$ or $P(A|B) = P(A)$, otherwise, A and B are *dependent*. If in an experiment the events A and B can both occur, then

$$P(A \cap B) = P(A)P(B|A)$$

We can also write:

$$P(A \cap B) = P(B \cap A) = P(B)P(A|B)$$

Two events A and B are independent if and only if

$$P(A \cap B) = P(A)P(B)$$

If the events $B_1, B_2, \ldots, B_k$ constitute a partition of the sample space S such that $P(B_i) \neq 0$ for $i = 1, 2, \ldots, k$, then for any event A of S,

$$P(A) = \sum_{i=1}^{k} P(B_i \cap A) = \sum_{i=1}^{k} P(B_i)P(A|B_i)$$

Often, the observations generated by different statistical experiments have the same general type of behavior. Since random variables associated with these experiments can be described by the same probability distribution and by a single formulae. The probability distribution can be described as *discrete* or *continuous* depending on whether they define probabilities associated with discrete random variables (i.e, if the set of its possible outcomes is countable) or continuous random variables (which can take values on a continuous scale). For example, suppose you flip a coin two times. The statistical experiment can have four possible outcomes: HH, HT, TH, and TT where H is Head

and T is Tail. Let the random variable X represent the number of Heads that result from the experiment. The random variable X can only take on the values 0,1, or 2. So, it is a discrete random variable.

The probability distribution of the discrete random variable X is a function $f(x)$ if, for every outcome x, we have:

1. $f(x) \geq 0$
2. $\sum_x f(x) = 1$
3. $P(X = x) = f(x)$

The cumulative distribution function $F(x)$ is defined as:

$$F(x) = P(X \leq x) = \sum_{t \leq x} f(t), \ for \ -\infty < x < \infty$$

The mean or expected value of X is:

$$\mu = E(X) = \sum_x x f(x)$$

The variance of the random variable X is:

$$\sigma^2 = E[(x-\mu)^2] = \sum_x (x-\mu)^2 f(x) = E(x^2) - \mu^2$$

The probability distribution of the continuous random variable X defined over the set of real numbers $\Re$ is $f(x)$, if

1. $f(x) \geq 0$ for all $x \in \Re$
2. $\int_{-\infty}^{\infty} f(x)dx = 1$
3. $P(a < x < b) = \int_a^b f(x)dx$

The cumulative distribution function $F(x)$ of the continuous random variable X is defined as:

$$F(x) = P(X \leq x) = \int_{-\infty}^{\infty} f(t)dt, \ for \ -\infty < x < \infty$$

The mean or expected value of X is:

$$\mu = E(X) = \int_{-\infty}^{\infty} f(x)dx$$

The variance of the random variable X is:

$$\sigma^2 = E[(x-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)dx$$

A.1.1 Some Discrete Probability Distributions

The simplest of all discrete probability distributions is one where the random variable assumes each of its values with an equal probability. Such a probability distribution is called a *discrete uniform distribution*. If the random variable X assumes the values $x_1, x_2, \ldots, x_k$, with equal probabilities, then the discrete uniform distribution is given by:

$$f(x,k) = \frac{1}{k}, \quad x = x_1, \ldots, x_k$$

We note that the uniform distribution depends on the parameter k. The mean of the discrete uniform distribution is:

$$\mu = \frac{\sum_{i=1}^{k} x_i}{k}$$

Since

$$\mu = E(x) = \sum_{i=1}^{k} x_i f(x_i, k) = \sum_{i=1}^{k} \frac{x_i}{k} = \frac{\sum_{i=1}^{k} x_i}{k}$$

The variance is:

$$\sigma^2 = \frac{\sum_{i=1}^{k} (x_i - \mu)^2}{k}$$

Since:

$$\sigma^2 = E[(X-\mu)^2] = \sum_{i=1}^{k} (x_i - \mu)^2 f(x_i, k) = \sum_{i=1}^{k} \frac{(x_i - \mu)^2}{k} = \frac{\sum_{i=1}^{k} (x_i - \mu)^2}{k}$$

An experiment often consists of repeated trials each with two possible outcomes that may be labelled success or failure. A success/failure experiment is called a *bernoulli experiment* or *bernoulli trail*. The Bernoulli process must possess the following properties:

1. The experiment consists of n repeated trials. Each trial results in an outcome that may be classified as a success or a failure.

2. The probability of success, denoted by p, remains constant from trial to trial.

3. The repeated trials are independent.

The most obvious example of a Bernoulli trial is coin tossing, where success means heads and failure means tails. The number X of successes in a Bernoulli trial is called a *Binomial random variable*. the probability distribution of this discrete random variable is called the *binomial distribution*, and its parameters will be denoted by $b(x; n, p)$. Therefore, a Bernoulli trial can result in a success with probability p and a failure with

probability $q = 1 - p$. The probability distribution of the binomial random variable X, the number of successes in n in independent trails, is:

$$b(x; n, p) = \binom{n}{x} p^x q^{n-x}, \quad x = 0, 1, 2, \ldots, n$$

For example, suppose that the probability that a certain kind of component will survive a given shock test is $\frac{3}{4}$. the probability that exactly 2 of the next 4 components tested survive is:

$$b(2; 4, \frac{3}{4}) = \binom{4}{2} (\frac{3}{4})^2 (\frac{1}{4})^2$$

the mean of the binomial distribution is:

$$\mu = np$$

The variance of a binomial distribution is:

$$\sigma^2 = npq$$

A.1.2 Some Continuous Probability Distributions

One of the simplest continuous probability distributions is the *continuous uniform distribution*. This distribution is characterized by a density function that is "flat" and thus the probability is uniform in a closed interval, say $[A, B]$. We can illustrate the density function of the continuous uniform distribution random variable X on the interval $[A, B]$ by the function:

$$f(x; A, B) = \frac{1}{B - A}, \quad A \leq x \leq B$$
$$f(x; A, B) = 0, \quad Elsewhere$$

Due to the simple nature of the density function, probabilities are simple to calculate for the uniform distribution. However, note that the application of this distribution is based on the assumption that the probability of falling in an interval within $[A, B]$ is constant.

The mean of the uniform distribution is $\mu = \frac{A+B}{2}$ and the variance $\sigma^2 = \frac{(B-A)^2}{12}$.

One of the most important continuous probability distributions is the normal distribution. Its graph, called the normal curve, is the bell-shaped curve as shown in Fig. A.1, describes approximately many phenomena that occur in nature, industry, and research.

A continuous random variable X having the bell-shaped distribution of Fig. A.1 is called a *normal random variable*. The mathematical equation for the probability

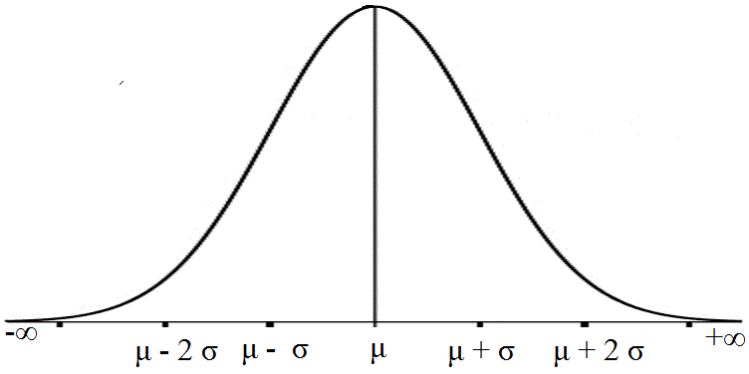

Figure A.1: The Normal curve.

distribution of the normal variable depends on two parameters: μ and σ, its mean and standard deviation. We denote the value of the density of X by $n(x;\mu,\sigma)$. The density function of the normal random variable X, with mean μ and variance σ^2 is:

$$n(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{(x-\mu)}{\sigma})^2}, \quad -\infty < x < \infty$$

The curve of any continuous probability distribution or density function is constructed in such a way that the area under the curve bounded by the two ordinates $x = x_1$ and $x = x_2$ equals the probability that the random variable X assumes a value between $x = x_1$ and $x = x_2$. Thus,

$$P(x_1 < X < x_2) = \int_{x_1}^{x_2} n(x;\mu,\sigma)dx = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{(x-\mu)}{\sigma})^2} dx$$

The difficulty encountered in solving integrals of normal density functions necessitates the tabulation of normal curve areas for quick reference. It is, however, a difficult task to set up a separate table for every conceivable value of μ and σ. Fortunately, we are able to transform all the observations of a random variable X to a new set of observations of a normal random variable Z with mean 0 and variance 1, which is denoted as the *standard normal distribution.*

This can be done by the transformation:

$$Z = \frac{X - \mu}{\sigma}$$

Note that if X is a binomial random variable with mean $\mu = np$ and variance $\sigma^2 = npq$, then the limiting form of the distribution of

$$Z = \frac{X - np}{\sqrt{npq}}$$

as n tends to ∞, is the standard normal distribution $n(Z; 0, 1)$. The normal distribution with $\mu = np$ and $\sigma^2 = np(1-p)$ provides a practical approximation to the binomial distribution when n is large and p is not close to 0 or 1.

The Normal distribution can be used to solve many problems in Management, Economics, Engineering, and Science there are still numerous situations that require different types of density functions such as the gamma and exponential distributions. The *gamma distribution* derives its name from the well-known gamma function:

$$\gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx, \ \forall \alpha > 0$$

The density function of the continuous random variable X which has a gamma distribution with parameters α and β is given by:

$$f(x) = \frac{1}{\beta^\alpha \gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}}, x > 0$$

$$f(x) = 0, \quad Elsewhere$$

where $\alpha > 0$ and $\beta > 0$.

The mean of the gamma distribution is $\mu = \alpha\beta$ and its variance is $\sigma^2 = \alpha\beta^2$. The special gamma distribution for which $\alpha = 1$ is called the exponential distribution. Therefore, the density function of the continuous random variable X which has an exponential distribution, with parameter β, is:

$$f(x) = \frac{1}{\beta} e^{-\frac{x}{\beta}}, x > 0$$

$$f(x) = 0, \quad Elsewhere$$

where $\beta > 0$.

The mean and variance of the exponential distribution are $\mu = \beta$ and $\sigma^2 = \beta^2$.

A.2 Sampling

A *population* consists of the totality of observations with which we are concerned, whether their number is finite or infinite such as groups of people, animals, etc. Each observation in a population is a value of a random variable X having some probability distribution $f(x)$. The statistician is interested in arriving at conclusions concerning a population when it is impossible or impractical to observe the entire set of observations that make up the population. That is why we must depend on a subset of observations from the population to help us make inference concerning that same population. Hence, we define the notion of sampling which is widely used for gathering information

about a population. A representative sample is one that has all the characteristics of the population from which it is drawn. A sample is a smaller (but hopefully a representative) collection of units from a population used to determine truths about that population. Any sampling procedure that produces inferences that consistently overestimate or consistently underestimate some characteristic of the population is said to be *biased*.

This process compromises several stages which are:

1. Defining the population of concern;
2. Specifying a sampling frame, ie.., as set of items or events possible to measure;
3. Specifying a sampling method for selecting items or events from the frame;
4. Determining the sample size;
5. Implementing the sampling plan;
6. Sampling and data collecting;
7. Data which can be selected (sampling frame).

We have two types of sampling. The first one is *probability sampling*, in which every unit in the population has a chance (greater than zero) of being selected in the sample. This probability can be accurately determined, when every element in the population has the same probability of selection. This is known as an equal probability of selection (EPS). The probability sampling methods include: simple random sampling (SRS), systematic sampling, stratified sampling, cluster sampling, and multistage sampling. The second one is *non probability sampling* which is any sampling method where some elements of the population have no chance of selection, or where the probability of selection cannot be accurately determined. It involves the selection of elements based on assumptions regarding the population of interest, which forms the criteria for selection. The selection of elements in this method is non random.

Next, we review the most popular methods of probability sampling that can be employed to draw conclusions about the population.

Simple random sampling. It is a subset of a statistical population in which each member of the subset has an equal probability of being chosen. A SRS is meant to be an unbiased representation of the population. An example of SRS is a group of 3 employees chosen out of 250 employees. In this case the population is all 250 employees and the sample is random because each employee has an equal chance of being chosen.

Systematic sampling. The steps that we need to follow in order to achieve a systematic random sample are:

1. Number the units in the population from 1 to N;

2. Decide on the sample size n that we need;

3. Calculate the interval size $k = \frac{N}{n}$;

4. Randomly select an integer between 1 and k;

5. Select every kth unit.

It is essential to know that the units in the population are randomly ordered, at least with respect to the characteristics we are measuring. Systematic random sampling is easy to implement and may be more precise than simple random sampling.

Stratified sampling. also sometimes called *proportional or quota random sampling*. It divides the population into homogenous subgroups (stratum) which share one common characteristic. Examples of stratum might be females and males. Once the stratum are defined, take a simple random sample in each subgroup. This method assures that you will be able to represent not only the overall population, but also the key subgroups of the population.

The main purpose in selecting random samples is to elicit information about the unknown population parameters. Any function of the random variables constituting a random sample is called a *statistic*. A statistic is a random variable that depends only on the observed sample, it must have a probability distribution. The probability distribution of a statistic is called *sampling distribution*. If $X_1, \ldots, X_n$ represent a random sample of size n, then the *sample mean* is defined by the statistic:

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

For instance, the sample mean $\hat{X}$ can be used to make an inference concerning the population mean μ. A measure of sample mean does not by itself give a clear indication of the nature of the sample. Thus, a measure of variability in the sample must be considered, i.e., the *sample variance*. The variability in the sample should display how the observations spread out from the average, i.e., the sample variance is defined as follows:

$$S^2 = \frac{\sum_{i=1}^{n} (X_i - \bar{X})^2}{n-1}$$

The *sample standard deviation* denoted by S, is the square root of the sample variance.

If we are sampling from a population with unknown distribution, the sampling distribution of $\bar{X}$ can be approximately normal with mean μ and variance $\frac{\sigma^2}{n}$ provided that the sample size is large. This is a result of the central limit theorem: If $\bar{X}$ is the

mean of a random sample of size n taken from a population with mean μ and finite variance σ^2, then the limiting form of the distribution

$$Z = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

as n tends to ∞, is the standard normal distribution $n(z; 0, 1)$.

Appendix B

Implementing Confidence Intervals and Hypothesis Testing in the $\mathcal{R}$ Language

We present the implementation details of the key building blocks of our experiments.

We used the $\mathcal{R}$ language and environment for statistical computing and graphics [R C13]. It is an environment which was developed at Bell Laboratories by John Chambers and colleagues. $\mathcal{R}$ is an environment within which many classical and modern statistical techniques have been implemented.

For each configuration $\langle \mu, \beta_h, \alpha_c, \vartheta \rangle$ and a given sample size n, we plan to perform 1000 random experiments by generating 1000 random configurations $\langle S_n, \beta_h, \alpha_c, \vartheta \rangle$.

For example, if we wish to generate a sample S_n of size n of values "0" and "1", we can use the Binomial distribution [Sam65] with parameters n and p which is a discrete probability distribution of the number of successes in a sequence of n independent yes/no experiments (Bernoulli trials), each of which yields success with probability p. A Bernoulli trial is a random experiment with exactly two possible outcomes, "success" and "failure", in which the probability of success is the same every time the experiment is conducted [Pap84]. The binomial distribution is frequently used to model the number of successes in a sample of size n drawn with replacement from a population.

We need to simulate random S_n that are Binomial. To do this, in $\mathcal{R}$, we call the *rbinom* function which has the following parameters:

1. The number of experiments. In our case, these correspond to the sample size n;
2. The number of observations per experiment. In our case, we only record one observation.

3. The probability of success p. In our case, this corresponds to μ.

So, to generate a sample of size 100, we call the following in $\mathcal{R}$:

```
S=rbinom(100,1,p) # generate a random sample of size 100
```

This will generate a vector of 0's and 1's of size 100. Each of the 0's and 1's is generated following the Bernoulli distribution with success probability p.

To compute a 95% Clopper-Pearson interval from a sample S of size n, we call in $\mathcal{R}$ the following commands:

```
zo=sum(S)
CI_o=binom.confint( x = zo, n, conf.level = 0.95, methods ="exact")
```

where:

- x is the number of successes in the binomial experiment.
- n is the number of independent trials in the binomial experiment.
- $conf.level$ is the level of confidence to be used in the confidence interval.
- $methods$ is the method to use to construct the interval. The "exact" corresponds to the Clopper-Pearson method.

The mean and the standard deviation of a sample can be computed in $\mathcal{R}$ using the following commands:

```
mn=mean(S)
stt<-sd(S)
```

The z-statistic —where the number of samples is n, β_h is denoted by $betah$ and ϑ by $theta$ —is computed in $\mathcal{R}$ as follows:

```
z <- (mn-(betah-theta))/(stt/sqrt(n))
```

For $\alpha_c = 0.95$, we reject the Null hypothesis $H_0 : \mu \geq \beta_h - \vartheta$ if $z < -1.645$ and accept $H_1 : \mu < \beta_h - \vartheta$. Otherwise, we accept the Null hypothesis H_0.