Using an SMT Solver for Interactive Requirements Prioritization

Francis Palma, Angelo Susi, Paolo Tonella
Fondazione Bruno Kessler
Software Engineering Research Unit
Trento, Italy
fpalma, susi, tonella@fbk.eu

Abstract

Requirements prioritization is a important activity in the initial stage of the software development process. It consists of finding an order relation among requirements, considering several requirements characteristics, such as stakeholders' preferences, technical constraints, implementation costs and user perceived value. An interactive approach has been proposed to the problem of prioritization based on Satisfiability Modulo Theory (SMT) techniques and pairwise comparisons. This approach resorts to interactive knowledge acquisition whenever the relative priority among requirements cannot be determined based on the available information. Synthesis of the final ranking is obtained via SMT constraint solving. The approach has been evaluated on a set of requirements from a real health-care project. Results show that it overcomes other interactive state-of-the art prioritization approaches in terms of effectiveness, efficiency and robustness to the decision maker errors.

KEYWORDS: Requirements Prioritization, Constraints Solving.

1 Introduction

Requirements prioritization is the process of specifying a total or partial order relation on a set of requirements under analysis. This process is of paramount importance in the first phases of the software development process when the scheduling of the software releases is planned and requirements are assigned to the releases, considering the stakeholders' expectations, implementation costs, time and other technical constraints.

Several approaches to requirements prioritization have been proposed in the last few years [1, 2, 3, 4, 5, 6, 7]. Some approaches [4, 7] rely on the definition of models that weight the attributes of the requirements (e.g. development costs, value for users, implementation effort) a priori, without any reference to the instances of requirements under analysis. Other approaches [1, 2, 3, 5, 6] allow the decision maker to formulate the ordering ex-post, on the basis of the characteristics of the specific set of requirements that is considered, without relying on predefined schemes.

Among the prioritization techniques used in the latter category, the Analytical Hierarchy Process (AHP) [8] is widely referenced. It exploits pairwise comparisons to extract the user knowledge with respect to the ranking of the requirements. The AHP method suffers scalability problems, since it defines the prioritization through user assessment of all the possible pairs of requirements. Hence, the number of pairwise comparisons asked to the user is quadratic with the number of requirements. In order to address the scalability problem of AHP, some techniques have been proposed [9, 10, 11]. They sub-sample the exhaustive set of pairwise comparisons, while trying to achieve comparable performance as AHP. IAHP (Incomplete Analytic Hierarchy Process) [10] and IGA (Interactive Genetic Algorithm) [11] are two of the most successful attempts made in this direction.

In this report we present all the results obtained through an ex-post approach that is based on the use of

Satisfiability Modulo Theory (SMT) techniques for the synthesis of a ranking from the constraints. Relying on pairwise preference elicitation, to extract relevant knowledge from the user, only when the available knowledge does not allow the disambiguation of the relative priority of a set of requirements, we used an SMT solver to compose the elicited preferences with the relevant ordering criteria induced by the attributes describing the requirements and encoded as constraints. The final objective is that of minimizing the user decision-making effort, increasing as much as possible the accuracy of the final requirements ranking. Specifically, pairs elicited from the user and initial constraints on the relative ordering of requirements define the set of constraints to be satisfied. Elicitation of stakeholders preferences and optimization are conducted at the same time and influence each other. In fact, the set of stakeholder constraints is specified incrementally, based on the unresolved priorities under the available constraints. Elicited constraints are iteratively composed with the constraints coming from other sources.

We evaluated this SMT approach comparing it with other state-of-the-art interactive prioritization techniques (IAHP and IGA in particular). Results indicate that SMT substantially outperforms them, while keeping the user effort (in terms of number of elicited pairs) acceptable. Moreover, we compared interactive and non-interactive prioritization, observing an increased performance when interaction with the decision maker takes place. We also evaluated the robustness of our method with respect to decision makers errors.

2 Background

2.1 Analytic Hierarchy Process (AHP)

The *Analytic Hierarchy Process (AHP)* was proposed by Saaty. It is one of the most accepted decision making methods [8]. Using AHP to prioritize software requirements involves comparing all unique pairs of requirements to determine which of the two is of higher priority, and to what extent. In a software project, comprising N user requirements, $\frac{N\times(N-1)}{2}$ pair-wise comparisons are asked to the decision maker. On one hand, AHP is a demanding method due to the dramatically increasing number of required pair-wise comparisons when the number of requirements grows; on the other hand, AHP is very trustworthy since the huge amount of redundancy in the pair-wise comparisons makes the process fairly insensitive to judgmental errors. Another advantage is that the resulting priorities are relative and based on a ratio scale, which allows for useful assessments of requirements.

Judgments and Comparisons: A judgment or comparison is the numerical representation of a relationship between two elements that shares a common parent. The set of all such judgments can be represented in a square matrix in which the set of elements is compared within itself. Each judgment represents the dominance of an element in the column on the left over an element in the row on top. It reflects the answers to two questions: which of the two elements is more important with respect to a higher level criterion and how strongly, using the 1-9 scale shown in Table 1. It is important to note that the lowest element is always used as the unit and the highest one is a multiple of that unit. From all the paired comparisons, priorities can be calculated and exhibited them on the upper right of the matrix. For a set of N elements in a matrix one needs $\frac{N\times(N-1)}{2}$ comparisons because there are N 1's on the diagonal for comparing elements with themselves and of the remaining judgments, half are reciprocals.

Intensity of Importance	Definition	Explanation
1	Equal importance	Two activities contribute equally to the objective
3	Moderate importance	Experience and judgment slightly favor one activity over another
5	Strong importance	Experience and judgment strongly favor one activity over another
7	Very string or demonstrated importance	An activity is favored very strongly over another; its dominance demonstrated in practice
9	Extreme importance	The evidence favoring one activity over another is of the highest possible order of affirmation
2, 4, 6, 8	For compromise between the above values	Sometimes one needs to interpolate a compromise judgment numerically because there is no good word to describe it

Table 1: Fundamental scales used for AHP (Karlsson and Ryan, 1997, p.70)

Scalability is a big problem for AHP. If the number of criteria is high and the number of alternatives is also high, the decision making process becomes complex and results in an enormous number of comparisons, which is not only tiresome but also inefficient from the user's perspective.

2.2 Interactive Genetic Algorithm

Often requirement analysts possess relevant knowledge about the relative importance and attributes of requirements. Interactive Genetic Algorithm (IGA) [11] can be used to produce a prioritized final requirements ordering which complies with user priorities, satisfies the technical constraints and takes into account the relative preferences elicited from the analyst. On a real case study (ACube [12]), it were shown that this approach improves non-interactive optimization that does not include the facility of elicited preferences, and that IGA can handle a number of requirements which is rather problematic for the state-of-the-art techniques. This solution belongs to the class of methods that is principally based on pair-wise comparison and exploits an IGA approach to minimize the number of pairs to be elicited from the stakeholder, whereas other state-of-the-art approaches do not have this advantage of extended interaction. Elicited pairs and initial constraints on the relative ordering of requirements define the complete fitness function. This function consists both of the disagreement between the requirements ordering encoded in an individual and the initial and elicited constraints.

Since elicitation and optimization are conducted at the same time during the evolutionary process, certainly they influence each other and a more specific uncommon characteristic of IGA is that the fitness function is constructed incrementally, being only partially known or even null at the beginning. Thus convergence is not trivially ensured by the optimization process. In [11], results indicate that IGA converges and improves the performance of GA (without interaction) by a considerable amount, in terms of disagreement with the *Gold Standard*, while keeping the user efforts (number of elicited pairs) within an acceptable upper limit.

2.2.1 Overall Proposed Process

The approach that leads to the goal of prioritization is performed through minimizing the disagreements between a total order of requirements holding priority values and different constraints that are embedded with them or that can be also generated or regenerated during the iterations. For the proposed approach, an *Interactive Genetic Algorithm* has been used to reach a targeted minimization by taking advantage of interactive feedback from the requirement engineers or stakeholders. This interaction feedback helps to define part of the fitness function that cannot be inferred from the existing information during the iterations.

Among the evolved populations, each individual represents an alternate solution candidate with a priority ordering. When individuals with the highest fitness values cannot be classified to extract the top rank one, as their fitness functions produce multiple equal valued orderings, user feedback is requested iteratively, to make the fitness function move downward, for further minimization. Highest fitness refers to the lowest disagreements w.r.t all constraints. The prioritization process terminates when either:

- a threshold disagreement is reached or
- the preferred time out is reached or
- a predefined total number of elicited pairs (the idea of elicitation process will be described formally later in this section) has been reached (but not crossed)

Threshold disagreement, time out and the total elicited pair; in the algorithm, these three are the baseline for termination during the evolutionary process, checking for minimum disagreement, execution time and count of elicited pairs respectively.

2.2.2 Algorithm

Algorithm 1 Compute Prioritized Requirements

Require: R: A set of requirements
Require: $ord_1..,...,..ord_k$: partial orders defining priorities and constraints upon R ($ord_i \subseteq R \times R$ defines a DAG)
Ensure: $< R_1..,...,..R_n >$: An ordered list of requirements
initialize *Population* with a set of ordered lists of requirements $\{Pr_i, ...\}$
elicitedPairs := 0
maxElicitedPairs := *MAX* (default = 100)
thresholdDisagreement := *TH* (default = 0)
topPopulationPerc := *PC* (default 1%)
eliOrd := $\emptyset$
for all Pr_i in *Population* **do**
  compute sum of *disagreement* for Pr_i w.r.t. $ord_1..,...,..ord_k$
end for
while *minDisagreement* $>$ *thresholdDisagreement* $\wedge$ *execTime* $<$ *timeOut* **do**
  sample *Population* with bias toward lower disagreement, e.g. using tournament selection
  sort *Population* by increasing *disagreement*
  if *minDisagreement* did not decrease during last G generations $\wedge$ there are ties in the *topPopulationPerc* of *Population* $\wedge$ *elicitedPairs* $<$ *maxElicitedPairs* **then**
    eliOrd := *eliOrd* $\cup$ elicit pairwise comparisons from user for ties
    increment *elicitedPairs* by the number of elicited pairwise comparisons
  end if
  mutate *Population* using the *requirement-pair-swap* mutation operator
  crossover *Population* using the *cut-head(tail)/fill-in-tail(head)* operator
  for all Pr_i in *Population* **do**
    compute sum of *disagreement* for Pr_i w.r.t. $ord_1..,...,..ord_k, eliOrd$
    update *minDisagreement*
  end for
end while
return Pr_{min}, the requirement list from *Population* with minimum *disagreement*

2.3 Incomplete Analytical Hierarchy Process

The two advantages which the AHP has over other multi-criteria methods are: (i) ease of use and (ii) the ability to handle inconsistencies in judgments. The experience with AHP supports Saaty's [8] claim that pairwise comparisons are somewhat *natural* i.e. individuals or groups quickly become comfortable with the pairwise comparison mechanism and find it easy to use.

The major drawback stays in individual or group decision process requiring significant amount of effort to complete all pair-wise comparisons. If we have 9 alternatives and 5 criteria, the total comparisons will be 190. In real-life problems these number of alternatives and criteria can be higher. This concern is handled by an almost functionally equivalent approach that requires less effort, called *Incomplete Analytic Hierarchy Process (IAHP).*

As P. T. Harker described it in [10], the steps used in the Incomplete Pairwise Comparison method are:

- Step 0: The decision maker provides N-1 judgments which form a connected graph if there are N alternatives.
- Step 1: Using the completed pairwise comparisons, derive the missing comparison by taking the geometric mean of intensities of a sample of random spanning trees. Calculate the weight matrix.
- Step 2: Calculate the derivatives of the weight matrix with respect to the missing matrix elements and select the next question by means of some mathematical derivations.
- Step 3: If this question meets the appropriate stopping criteria then stop; else elicit this comparison and return to step 1.

There are three possible ways by which a decision maker can stop the comparison process:

- The decision maker decides whether or not to continue with the further questioning; or,
- If the maximum absolute difference in the attribute weights from one question to another is $\leq \alpha\%$, where α is a given constant; or,
- The comparisons will continue to be made until one is sure that the ordinal rank will not be reversed.

To explain the last stopping criteria, we can state that the weights are cardinal rankings of the alternatives, which creates an ordinal ranking for them as well. But if the decision maker answers too many or more questions, the cardinal ranking in weights may be slightly altered but the ordinal ranking would remain the same. So, the third stopping rule states that the next question is going to be asked only if the ordinal ranking could be reversed. Using IAHP, substantial effort (i.e. time) saving can be achieved w.r.t. classical AHP but still inefficient to handle large set of requirements.

3 Proposed SMT-based Interactive approach

3.1 Interactive SMT-based prioritization

The prioritization approach we propose aims at minimizing the disagreement between a total order of prioritized requirements and the various constraints that are either encoded with the requirements or that are expressed iteratively by the user during the prioritization process. We use SMT solvers to achieve such a minimization, taking advantage of interactive input from the requirement engineer whenever the solver produces more than one prioritized list of requirements having the same disagreement with the available constraints. The prioritization process terminates when a unique solution at minimum disagreement is found or the maximum allowed number of pairwise comparisons is reached.

Constraints upon requirements can be represented by means of a *constraint* graph. In a constraint graph, an edge between two requirements indicates that, according to the related dependency or priority, the requirement associated with the source of the edge should be implemented before the target requirement. Edges may be weighted, to actually quantify the strength or importance of each constraint. When encoded in the input language of the SMT solver, such constraints will be retractable constraints which are given the weight labeling the edge (or 1 if no weight is given). An infinite weight is used for constraints that must necessarily hold in the final ordering of the requirements. These will be non-retractable constraints for the SMT solver.

It is possible to encode the requirements prioritization problem as a MAX-SAT problem. An SMT solver will find an integer assignment which maximizes the weight of the retractable constraints that are satisfied by the solution (minimum cost of unsatisfied constraints). The encoding is described formally in the next section. Intuitively, it consists of an assignment of positions (integers between 1 and N, for N requirements) to the components of an integer array x of size N. The assignment must define a permutation of the positions, hence there should be no repetition of positions. This is easily encoded as a set of inequalities between pairs of positions ($x[i] \neq x[j] \quad \forall\, i \neq j$). The other constraints (from the constraint graphs) are encoded as inequalities (e.g., $x[1] < x[4]$ and $x[1] < x[5]$ for the two edges at the top of the constraint graph Prio). The solutions to the MAX-SAT problem produced by the SMT solver are all requirement orderings that violate the minimum number of retractable constraints (e.g., Deps and Prio), or have minimum cost of violation, in case different weights are given to different constraints.

3.2 Algorithm

Algorithm 2 Compute prioritized requirements

Require: R: set of requirements
Require: $ord_1, ..., ord_k$: partial orders defining constraints upon R ($ord_i \subseteq R \times R$)
Ensure: $\langle R_1, ..., R_n \rangle$: ordered list of requirements

```
Solutions := ∅
elicitedPairs := 0
maxElicitedPairs := MAX_ELI_PAIRS (default = 100)
eliOrd := ∅
while |Solutions| ≠ 1 ∧ elicitedPairs < maxElicitedPairs do
    Solutions := MAX_SAT(ord_1, ..., ord_k, eliOrd)
    if |Solutions| ≠ 1 then
        eliOrd := eliOrd ∪ elicit pairwise comparisons from user for disagreement pairs up to maxElicitedPairs
        increment elicitedPairs by the number of elicited pairwise comparisons
    end if
end while
return Pr_min, a requirement list from Solutions with minimum disagreement
```

4 Implementation

We implemented the algorithm of SMT-based interactive approach in C++ under Eclipse IDE, using the C SMT solver library provided with the SMT tool Yices. And IAHP, IGA were implemented in Java.

The initial implemented version of the software and all the experimental data are available online at the

URL `http://selab.fbk.eu/smt-req-prio`.

5 Experimental Settings

The three main components of an experimental design are (i) factors, (ii) levels and (iii) response.

Factors refer to the inputs to the process and can be classified as either controllable or uncontrollable variables. In our case main factors are (i) Applied algorithm and (ii) User model. There are also other minor factors i.e. number of elicited pair, total execution time, minimum threshold disagreement etc.

Levels or settings of each factor i.e. the value for those factors that are defined within the environment and going to be used during the execution.

Response or output of the experiment. In our case, mainly the disagreement w.r.t. the *Gold Standard*.

5.1 Experimental Design: The factors to be tested

(i) *Applied Algorithm*, for our experimental design we used two other interactive prioritization algorithms. We also want to compare the resulting disagreement w.r.t. GS of the final ordering using both interactive & non-interactive approaches.

(ii) *User Model*, in our experiments we have shown both the cases where there is an ideal user (i.e. user makes no error) and the user that frequently gives wrong answers. So, having different settings for the user model will help us to evaluate our algorithm under different user error conditions.

The other minor factors that are merely considered as an important part of the optimization process are:

(a) *Maximum Elicited Pairs*, for each target objective we will have a maximum number of elicited pair, the algorithm terminates when it reaches this value.
(b) *Threshold Minimum Disagreement*, a minimum disagreement value will be used as threshold value.
(c) *Time Out*, the total *timeOut* set for running the evolutionary part of the algorithm. After the specified *timeOut* is reached the evolution process will terminate (for IGA).
(d) *Top Population Percentage*, if we have N individuals in the population and *topPopulationPerc* is set to k% then N×k% will be evaluated for further discrimination using interaction with the user when there are ties.

5.2 Experimental Preparation: The levels of the factors

(i) *Applied Algorithm*: The algorithm that we apply is the main factor for us i.e. use of different interactive algorithms produce different levels of disagreement.

Factor	Levels	Category
Algorithm	SMT-based IGA IAHP	Interactive, *Domain Knowledge*, *User Knowledge* through pairwise comparisons

Table 2: Levels for the factors used in the experiments

(ii) *User Model*: We have conducted experiments with all four macro scenarios with a different number of pairs elicited from the user and with variations of user error rates. More specifically, for all macro scenarios we experimented with 25, 50 & 100 elicited pairs and error rates between 0% and 20%. Table 3 reflects the user and error model we had in our case study (numbers remain same for all macro scenarios i.e. ALL, FAL, ESC & MON):

A Tot. Eli Pair	B Err Rate	C Wrong Eli Pair
25	5%	2
	10%	3
	20%	5
50	5%	3
	10%	5
	20%	11
100	5%	5
	10%	10
	20%	20

Table 3: Numerical representation of user error model used for simulation in SMT, IGA, IAHP

In Table 3, column A represents the total number of elicited pairs we experimented with. Column B represents error the rates we used. Column C represents the calculated number of elicited pairs that were wrongly answered by user.

The other levels of some minor factors are below:

(a) *Maximum Elicited Pairs*, the default value is 100; we have experimented all the macro scenarios with 25, 50 and 100 elicited pairs. We then terminate the evolution process and measure the disagreement with the *Gold Standard*. The primary objective of the diversity in setting the number of maximum elicited pair is to see the fluctuations on the disagreement w.r.t. Gold Standard i.e. how close we can reach in terms of orderings.
(b) *Threshold Minimum Disagreement*, default value is 0; For our experiments, the threshold value was set to zero to reach the closest ordering to GS.
(c) *Time Out*, the total time out set for our experiments was between 480 and 1080 seconds depending on the number of requirements and number of elicited pairs (for IGA) and for SMT & IAHP, they terminates whenever they reached maximum elicited pairs.
(d) *Top Population Percentage*, default value is 1% for the *topPopulationPerc*.

The structure and layout of experimental runs, or conditions:
For each macro scenarios, we run experiments multiple times (for IGA) depending on the total execution time it takes for each run (we reduced the number of runs if it exceptionally consumes more time i.e. if it takes 1800 seconds or more per run).

5.3 Execution

Since IGA is non deterministic, we replicated each experiments a minimum number of 10 to maximum of 20 times with a *timeOut* between 600 & 1080 seconds. We varied this execution time due to variability in elicited pairs, the more pairs we elicit, in practice the more operational time it consumes. So, for the proper judgment among the execution settings we assumed a range of 600, 840 and 1080 execution time (in seconds) while eliciting 25, 50 & 100 pairs respectively from the user (for IGA). We then compute disagreements and average distances after the optimization process terminates. We simulate the artificial user who automatically responds according to the GS in case of error-free settings, and just in the reverse order in case of erroneous settings i.e. when setting the p_c value greater than zero, the user will make wrong elicitation with the probability of p_c. Finally, we do the box-plots for all the results we had, to have a clear comparable view with other interactive approaches.

6 Results

6.1 Results along with IGA, IAHP and non-Interactive SMT

Table 4 in Section 6.1.1, Table 5 in Section 6.1.2, Table 6 in Section 6.1.3 and Table 7 in Section 6.1.4 represent the four summary results that we obtained for four macro scenario ALL, MON, FALL & ESC respectively.

6.1.1 Results for ALL Macro Scenario

		SMT		SMT($p_e = 5\%$)			SMT($p_e = 10\%$)			SMT($p_e = 20\%$)			IGA		IAHP	
Eli.pairs	Actual	Dis	AD	Err	Dis	AD	Err	Dis	AD	Err	Dis	AD	Dis	AD	Dis	AD
25	25	92	3.18	1	102	3.43	3	102	3.42	5	101	3.39	124	4.06	478	13.60
50	50	90	3.14	3	92	3.16	5	95	3.22	10	98	3.31	120	3.82	208	6.30
100	84	73	2.78	4	75	2.86	8	79	2.94	17	85	3.10	114	3.69	187	5.75

Table 4: Comparison table of disagreement and average distance among SMT, IGA & IAHP at different user error rates for ALL scenario

6.1.2 Results for MON Macro Scenario

		SMT		SMT($p_e = 5\%$)			SMT($p_e = 10\%$)			SMT($p_e = 20\%$)			IGA		IAHP	
Eli.pairs	Actual	Dis	AD	Err	Dis	AD	Err	Dis	AD	Err	Dis	AD	Dis	AD	Dis	AD
25	25	14.5	1.24	1	14.5	1.24	3	15	1.33	5	15	1.33	25	2.10	43	3.29
50	31	10	0.95	2	10	0.95	3	10	0.95	6	10	0.95	20	1.67	29	2.19
100	31	10	0.95	2	10	0.95	3	10	0.95	6	10	0.95	15	1.33	16	1.24

Table 5: Comparison table of disagreement and average distance among SMT, IGA & IAHP at different user error rates for MON scenario

6.1.3 Results for FALL Macro Scenario

		SMT		SMT($p_e = 5\%$)			SMT($p_e = 10\%$)			SMT($p_e = 20\%$)			IGA		IAHP	
Eli.pairs	Actual	Dis	AD	Err	Dis	AD	Err	Dis	AD	Err	Dis	AD	Dis	AD	Dis	AD
25	25	5	0.38	1	6	0.46	3	8	0.54	5	9	0.62	20	1.42	46	2.85
50	42	0	0.00	2	0	0.00	4	0	0.00	8	0	0.00	17	1.08	55	3.42
100	42	0	0.00	2	0	0.00	4	0	0.00	8	0	0.00	11	0.77	28	1.73

Table 6: Comparison table of disagreement and average distance among SMT, IGA & IAHP at different user error rates for FALL scenario

6.1.4 Results for ESC Macro Scenario

		SMT		SMT($p_e = 5\%$)			SMT($p_e = 10\%$)			SMT($p_e = 20\%$)			IGA		IAHP	
Eli.pairs	Actual	Dis	AD	Err	Dis	AD	Err	Dis	AD	Err	Dis	AD	Dis	AD	Dis	AD
25	25	8	0.65	1	11	0.96	3	12	1.04	5	12.5	0.96	20	1.35	42	2.78
50	34	5	0.43	2	5	0.43	3	5	0.43	7	5	0.43	19	1.35	37	2.48
100	34	5	0.43	2	5	0.43	3	5	0.43	7	5	0.43	13	1.05	21	1.43

Table 7: Comparison table of disagreement and average distance among SMT, IGA & IAHP at different user error rates for ESC scenario

6.2 Results for Research Question 1

RQ1 (Comparison) *Does the SMT-based method produce improved prioritizations compared to IAHP and IGA?*

We compare SMT-based interactive prioritization with the state of the art interactive requirement prioritization technique IAHP [10] and with our previous work IGA [11]. The output of SMT-based prioritization is compared with the output of the IAHP and IGA algorithms, at equal number of pairs elicited from the user. We vary such number from low to high values, in order to investigate the regions where one approach is superior to the other (i.e., the degree of information incompleteness each approach can accommodate).

To assess the performance of IGA and IAHP we use two main metrics: disagreement with GS and average distance from the position of each requirement in the GS. The latter metrics is highly correlated with the former, but it has the advantage of being more easily interpretable than disagreement. In fact, disagreement involves a quadratic number of comparisons (each pair of requirements w.r.t. GS order), hence its absolute value is not straightforward to understand and compare. On the contrary, the distance between the position of each requirement in the prioritization produced by our algorithm and the position of the same requirement in the GS gives a direct clue on the number of requirements that are incorrectly positioned before or after the requirement being considered.

6.2.1 Results for Research Question 1: ALL Macro Scenario

Macro scenario	Parameters	Value
ALL (49 req)	targetAlgorithm	SMT, IGA, IAHP
	measurement	Disagreement, Average Distance
	maxElicitedPairs	25, 50 & 100
	errorPerc	0% error rate

Table 8: Experimental settings for the RQ1: ALL Macro Scenario

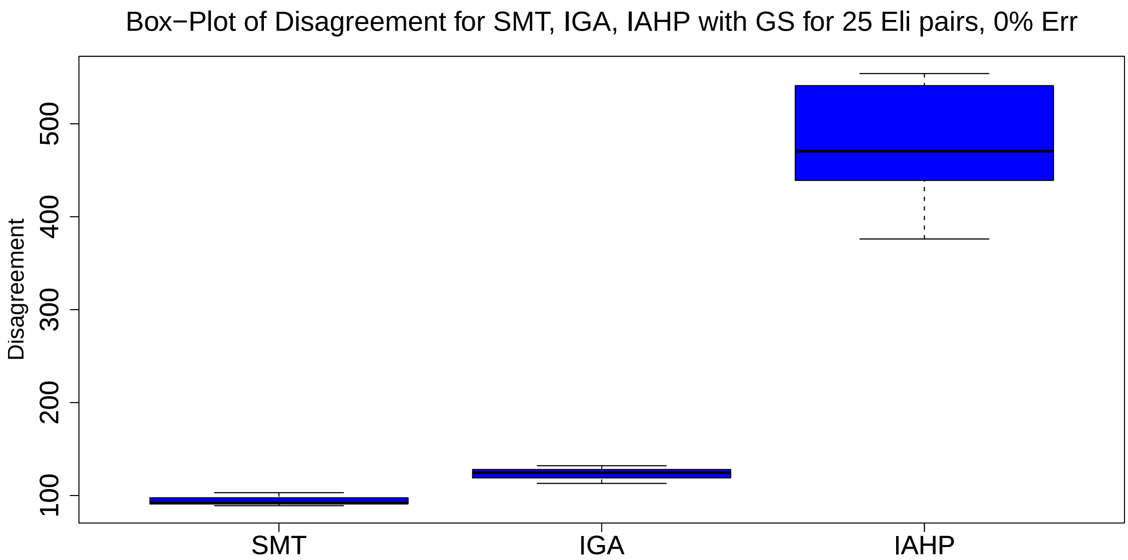

Figure 1: Box-Plot of Disagreement for SMT, IGA and IAHP with GS for 25 Eli pairs, 0% user error, 49 Req

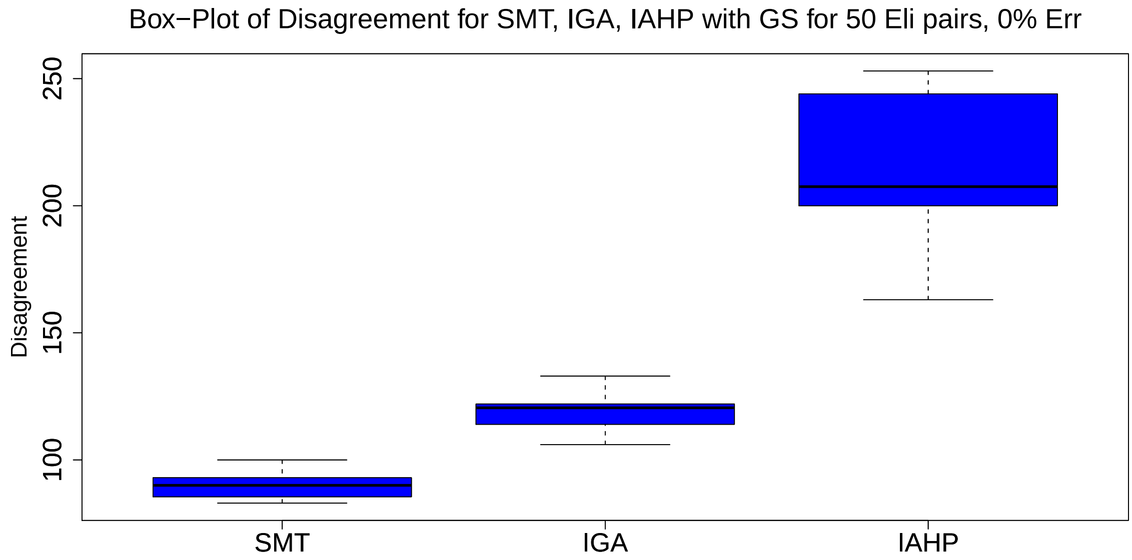

Figure 2: Box-Plot of Disagreement for SMT, IGA and IAHP with GS for 50 Eli pairs, 0% user error, 49 Req

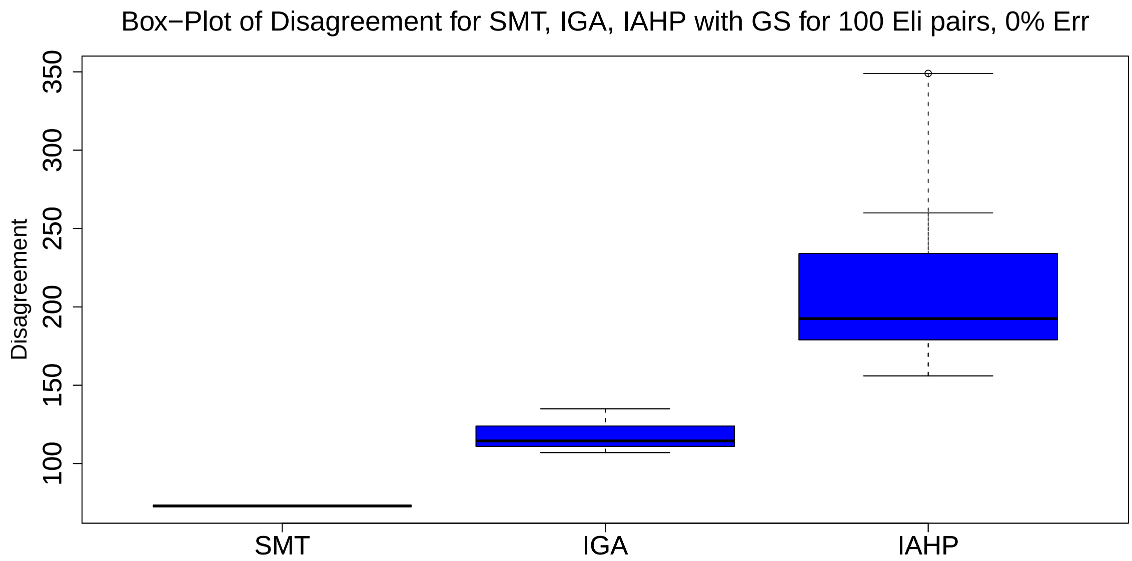

Figure 3: Box-Plot of Disagreement for SMT, IGA and IAHP with GS for 100 Eli pairs, 0% user error, 49 Req

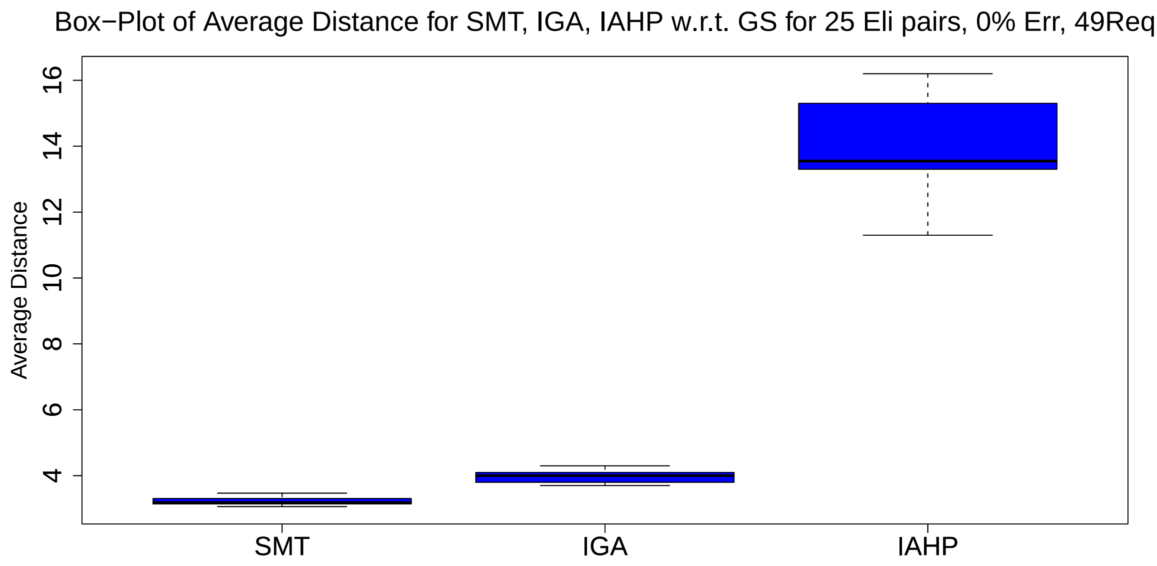

Figure 4: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 25 Eli pairs, 0% user error, 49 Req

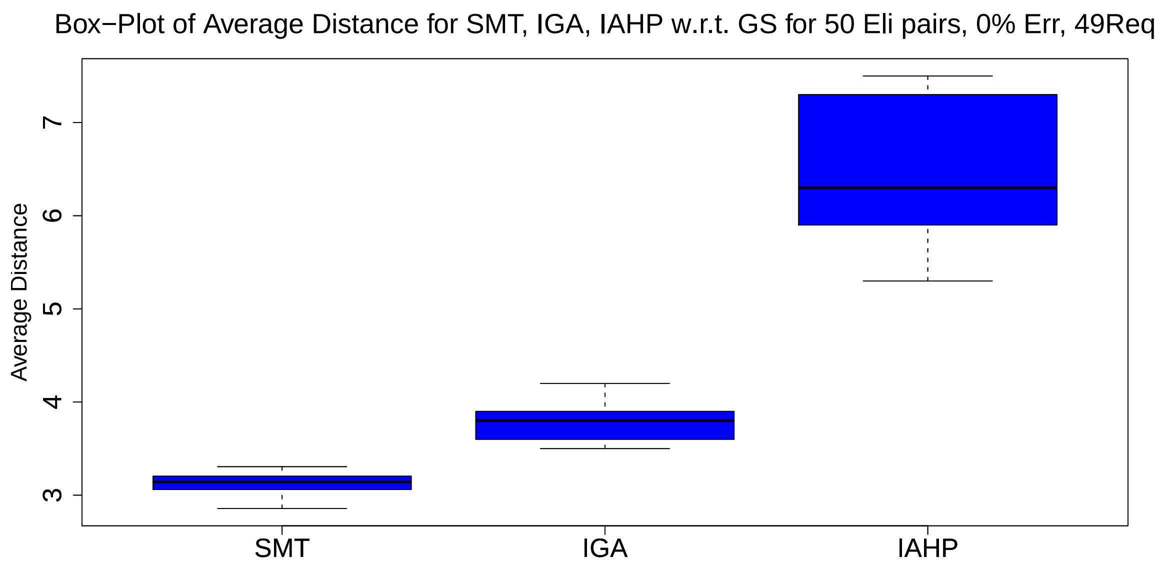

Figure 5: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, 0% user error, 49 Req

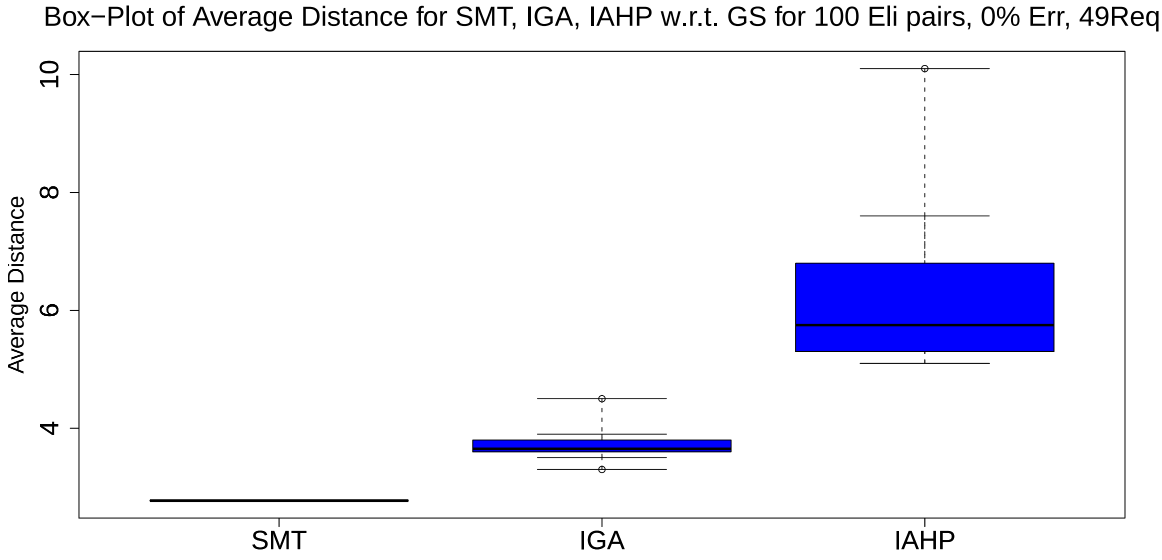

Figure 6: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, 0% user error, 49 Req

6.2.2 Results for Research Question 1: ESC Macro Scenario

Macro scenario	Parameters	Value
ESC (23 req)	targetAlgorithm measurement maxElicitedPairs errorPerc	SMT, IGA, IAHP Disagreement, Average Distance 25, 50 & 100 0% error rate

Table 9: Experimental settings for the RQ1: ESC Macro Scenario

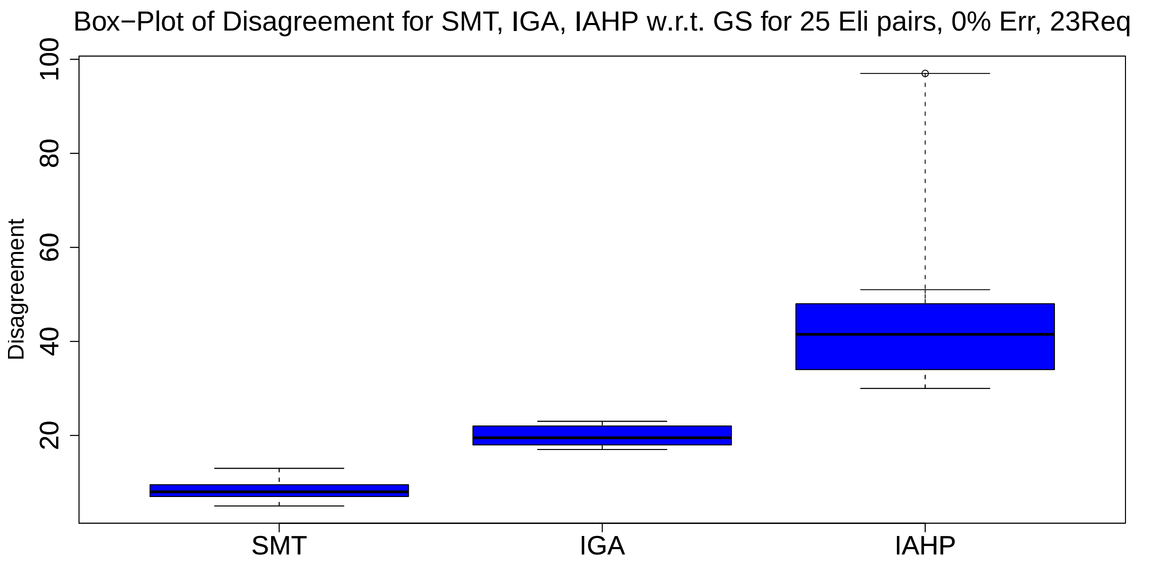

Figure 7: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 25 Eli pairs, 0% user error, 23 Req

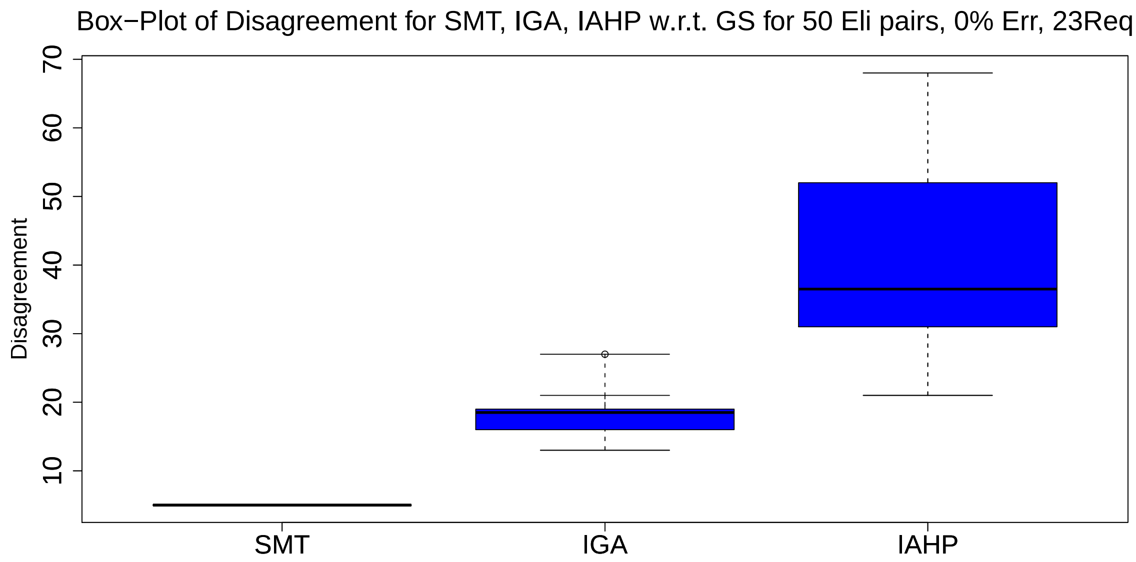

Figure 8: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, 0% user error, 23 Req

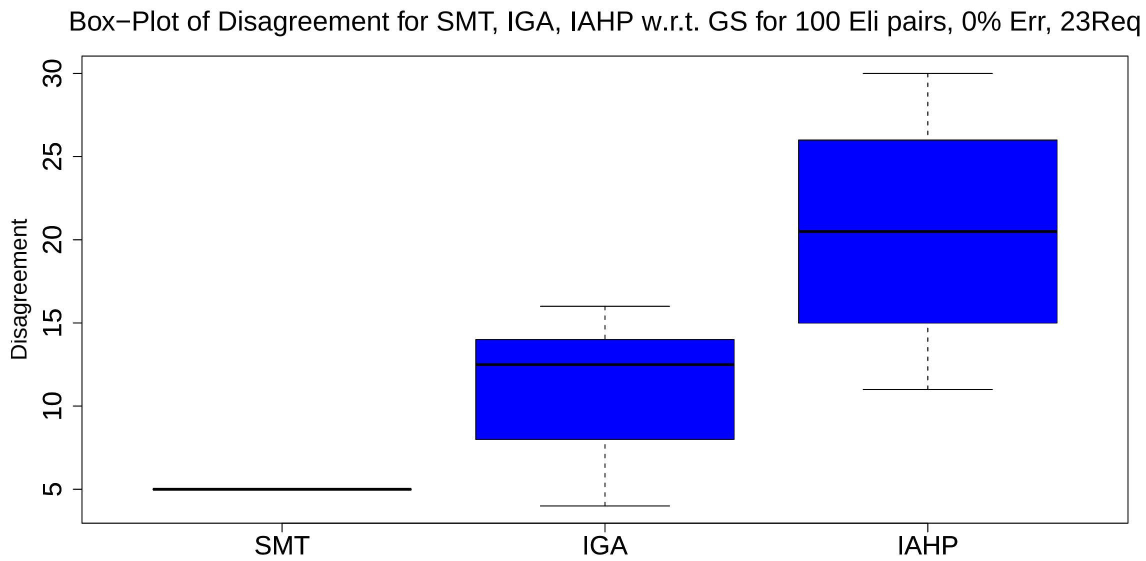

Figure 9: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, 0% user error, 23 Req

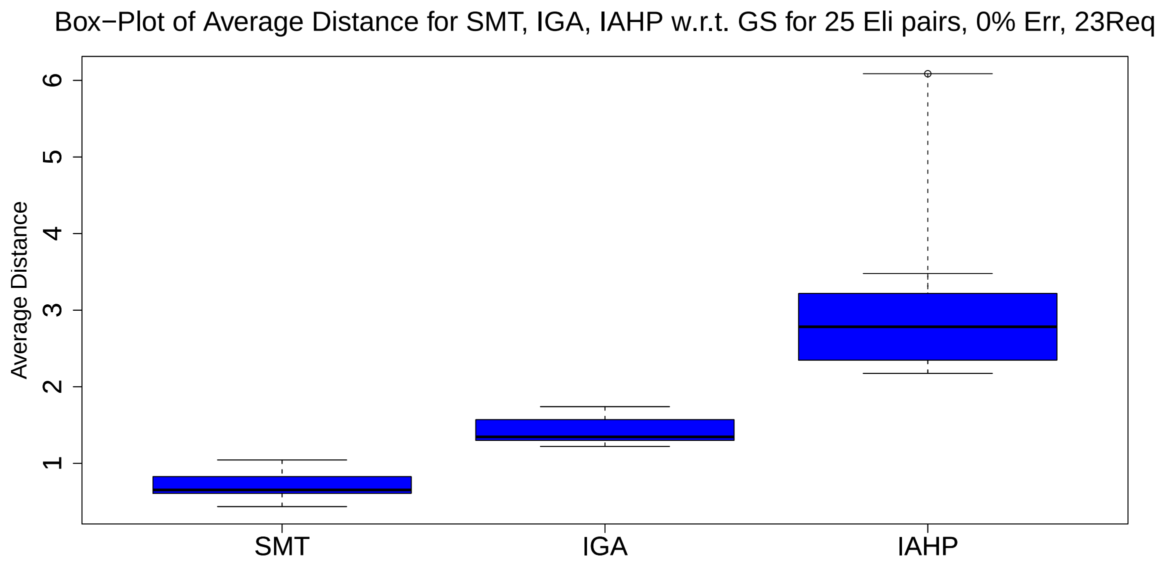

Figure 10: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 25 Eli pairs, 0% user error, 23 Req

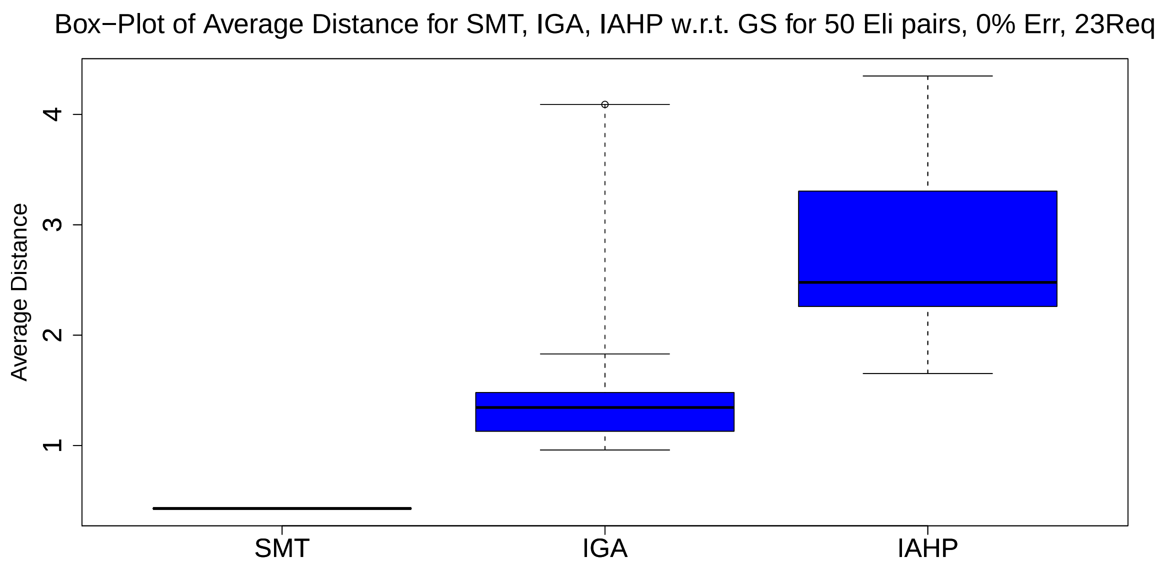

Figure 11: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, 0% user error, 23 Req

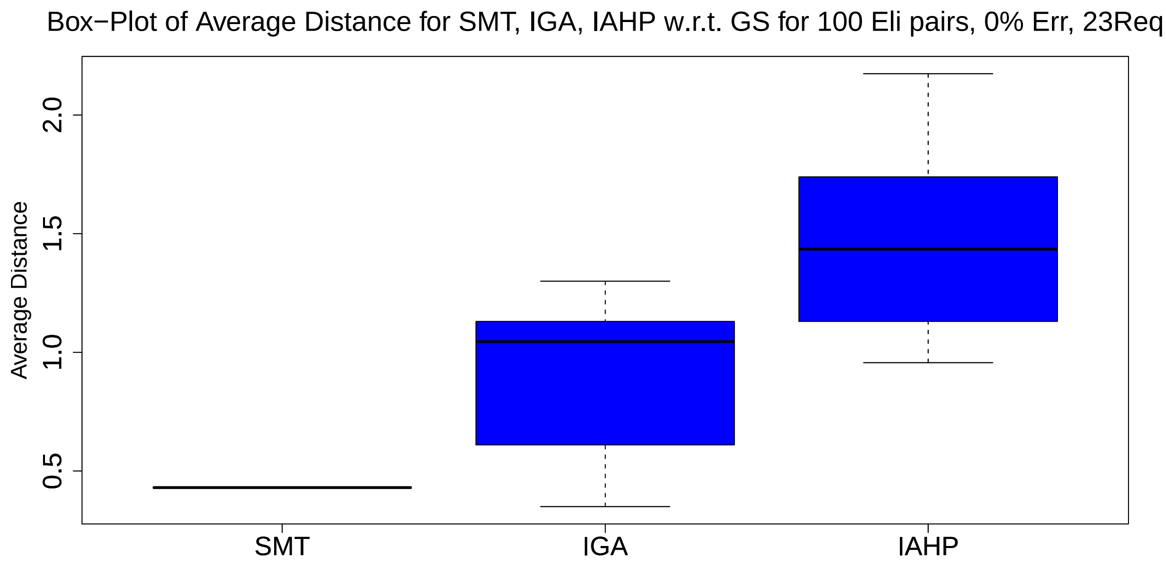

Figure 12: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, 0% user error, 23 Req

6.2.3 Results for Research Question 1: FALL Macro Scenario

Macro scenario	Parameters	Value
FALL (26 req)	targetAlgorithm measurement maxElicitedPairs errorPerc	SMT, IGA, IAHP Disagreement, Average Distance 25, 50 & 100 0% error rate

Table 10: Experimental settings for the RQ1: FALL Macro Scenario

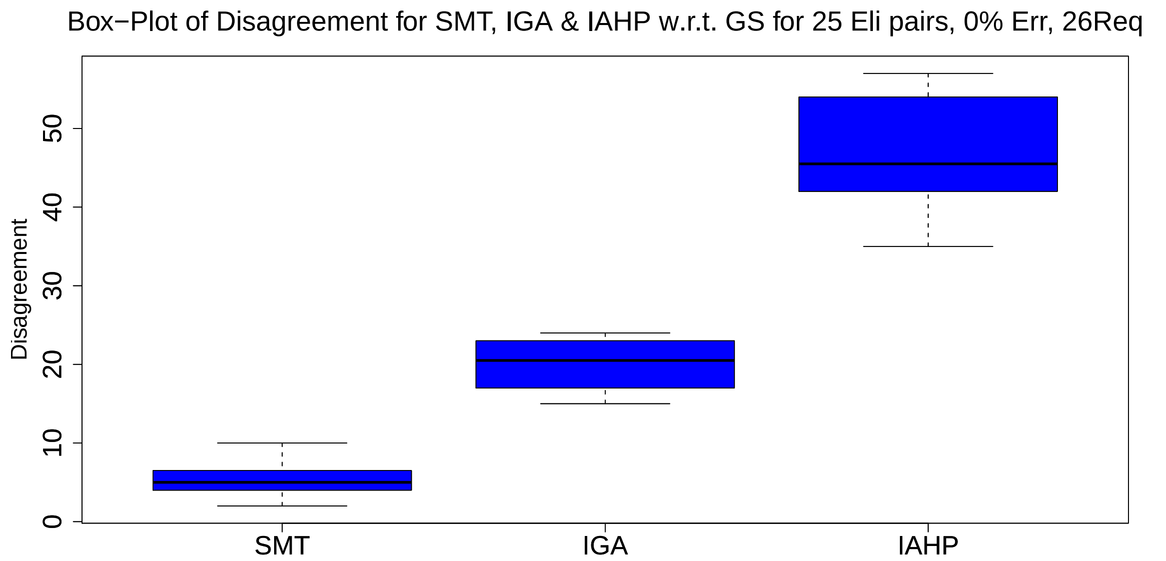

Figure 13: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 25 Eli pairs, 0% user error, 26 Req

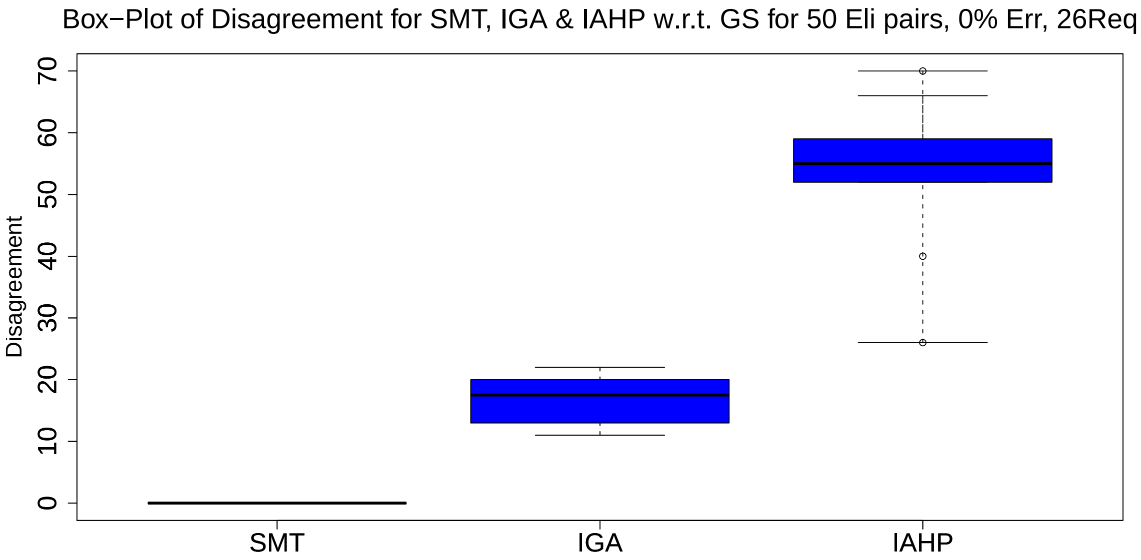

Figure 14: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, 0% user error, 26 Req

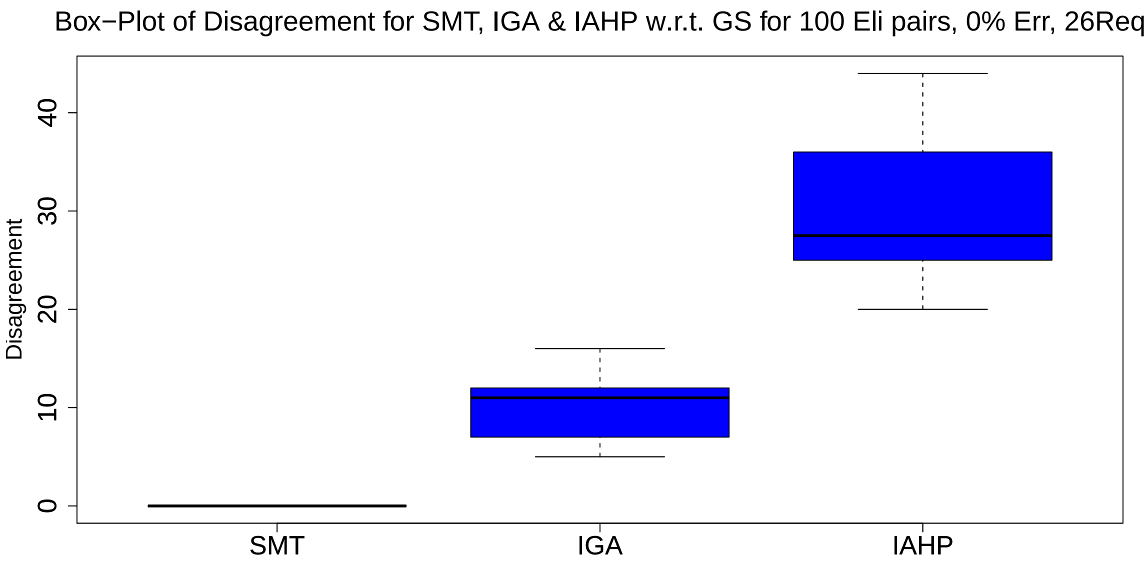

Figure 15: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, 0% user error, 26 Req

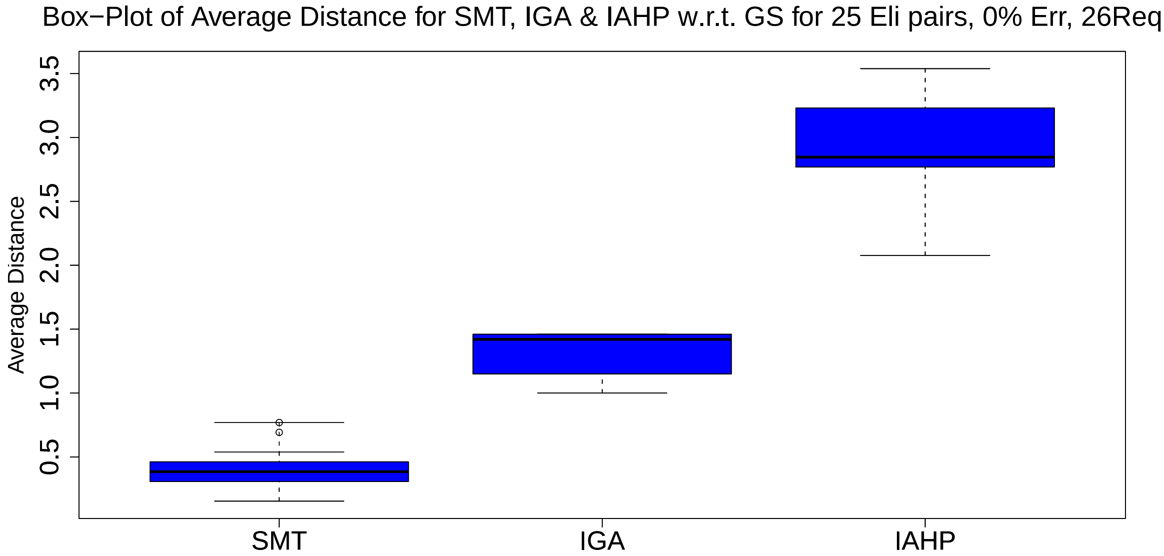

Figure 16: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 25 Eli pairs, 0% user error, 26 Req

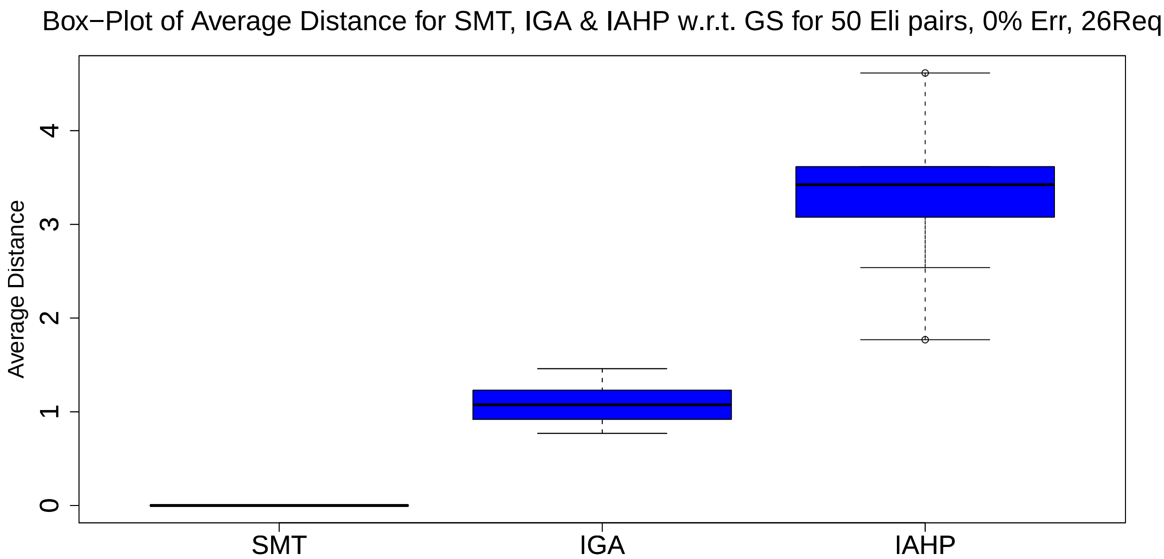

Figure 17: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, 0% user error, 26 Req

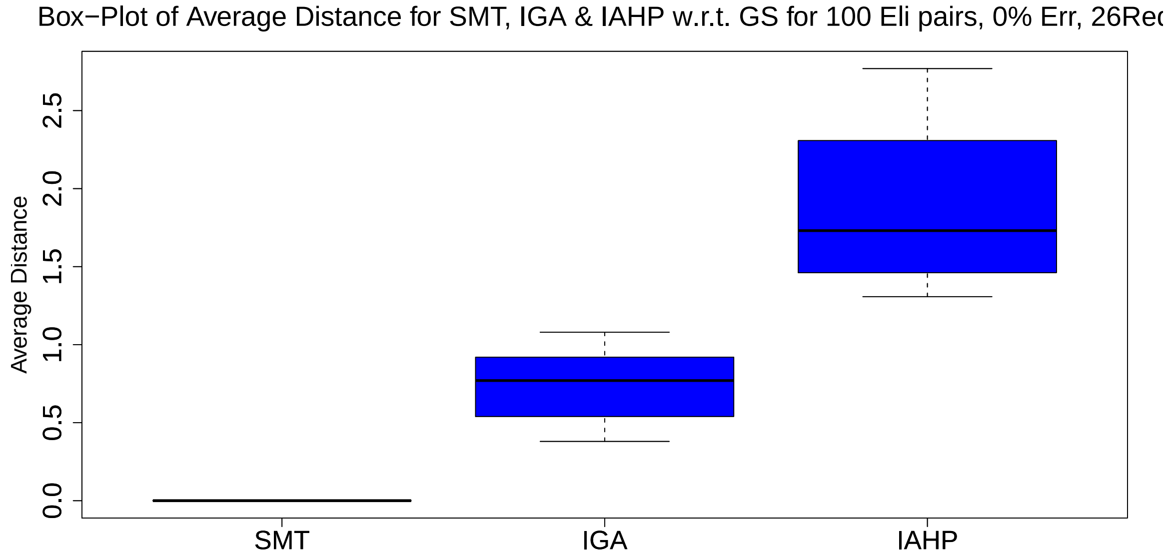

Figure 18: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, 0% user error, 26 Req

6.2.4 Results for Research Question 1: MON Macro Scenario

Macro scenario	Parameters	Value
MON (21 req)	targetAlgorithm	SMT, IGA, IAHP
	measurement	Disagreement, Average Distance
	maxElicitedPairs	25, 50 & 100
	errorPerc	0% error rate

Table 11: Experimental settings for the RQ1: MON Macro Scenario

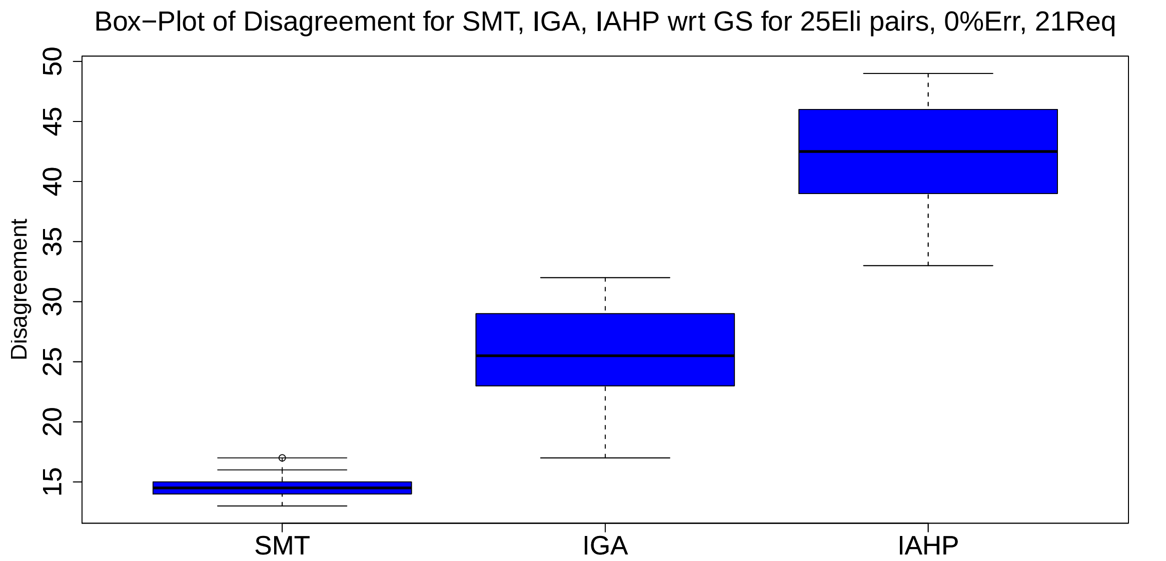

Figure 19: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 25Eli pairs, 0% user error, 21 Req

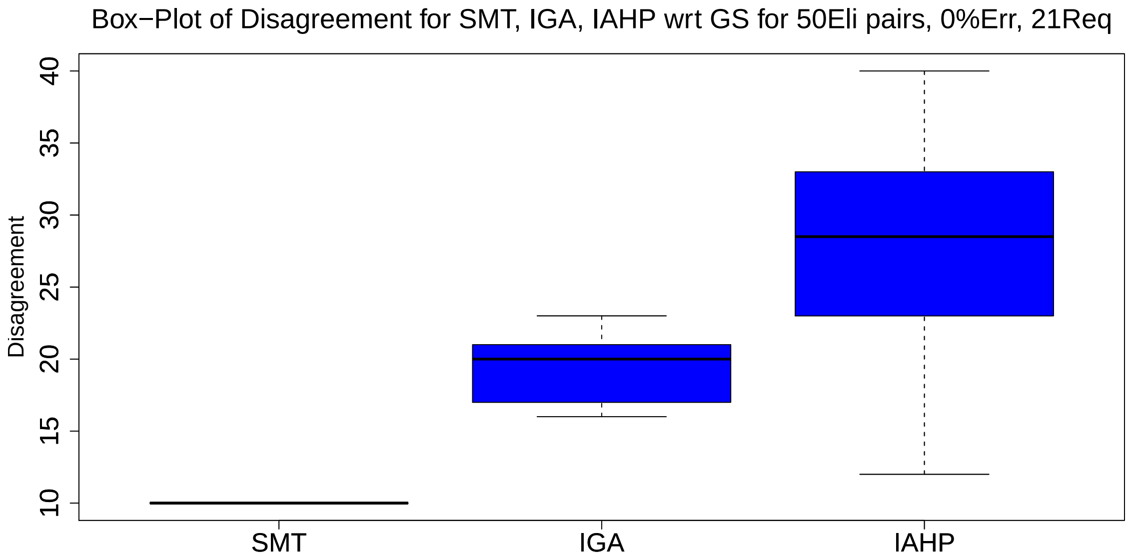

Figure 20: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 50Eli pairs, 0% user error, 21 Req

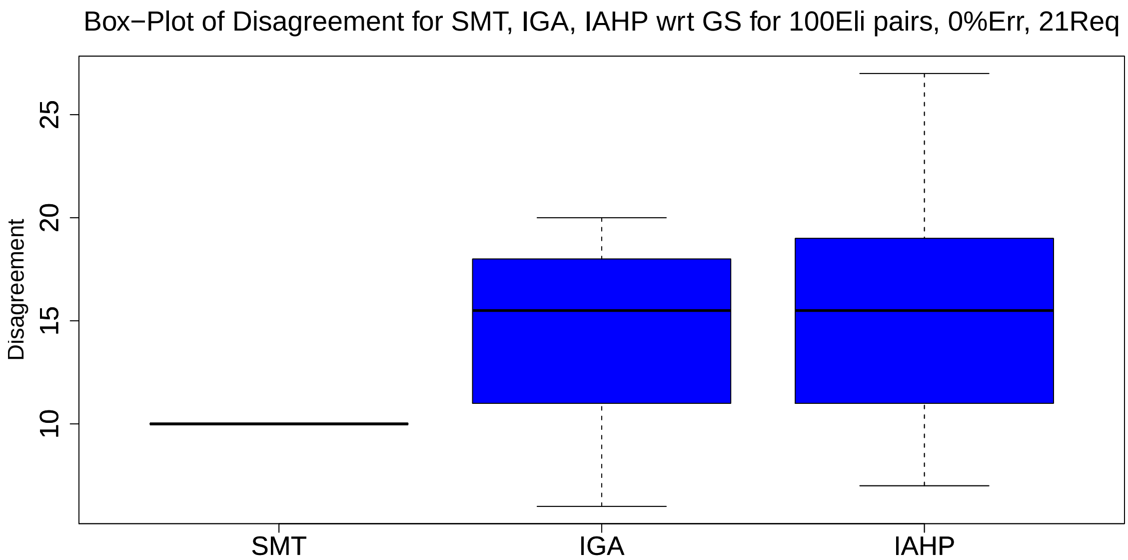

Figure 21: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 100Eli pairs, 0% user error, 21 Req

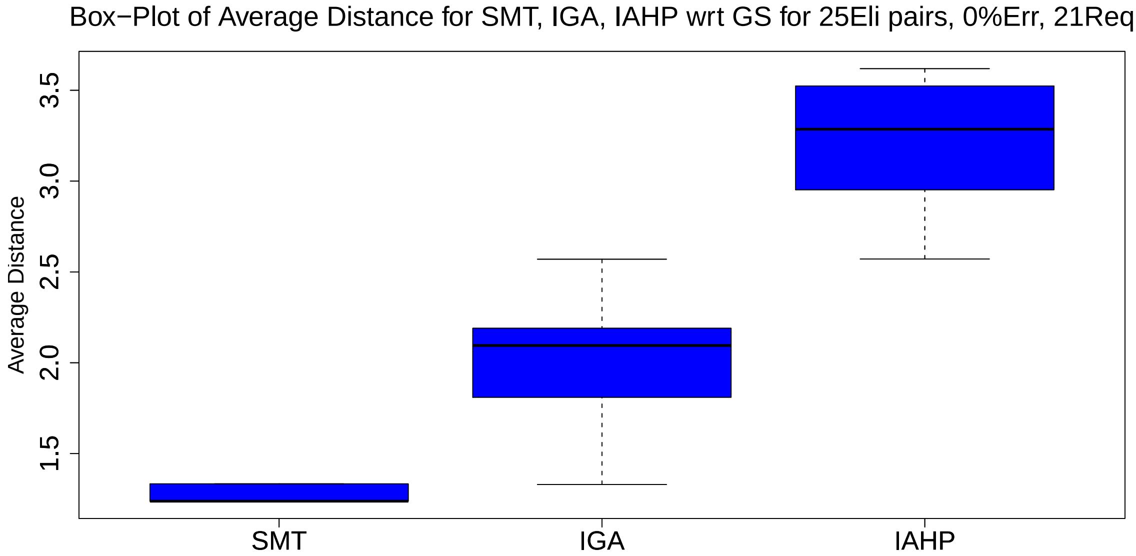

Figure 22: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 25Eli pairs, 0% user error, 21 Req

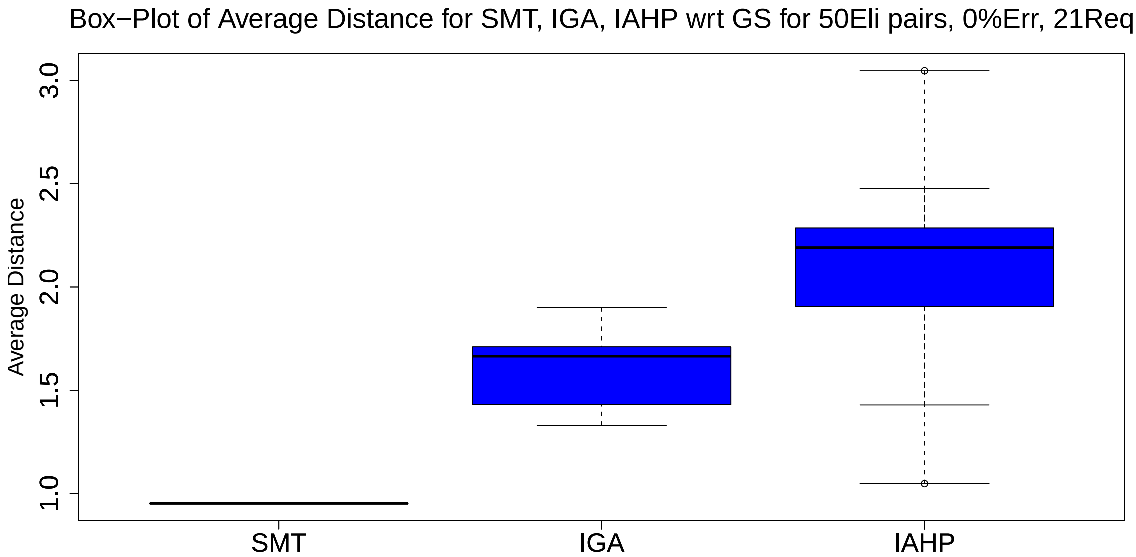

Figure 23: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 50Eli pairs, 0% user error, 21 Req

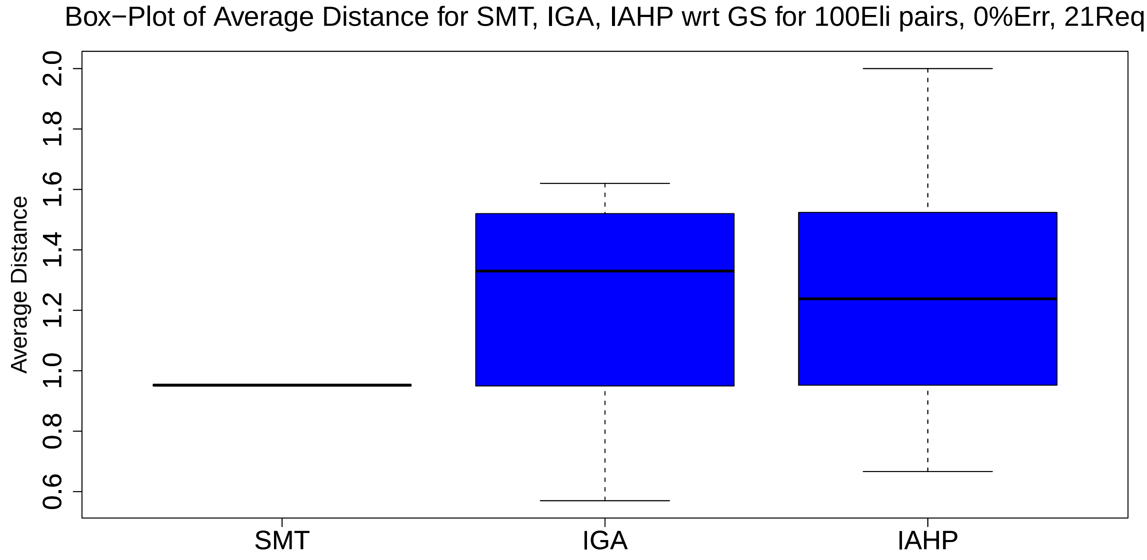

Figure 24: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 100Eli pairs, 0% user error, 21 Req

6.2.5 Statistical Comparison: Analysis of Variance (ANOVA)

Methods	Disagreement p-value	Avg. Distance p-value
SMT25, IGA25, IAHP25	<2.2e-16	<2.2e-16
SMT50, IGA50, IAHP50	<2.2e-16	<2.2e-16
SMT100, IGA100, IAHP100	9.8e-05	1.03e-04

Table 12: Analysis of variance (ANOVA) comparing SMT, IGA and IAHP for ALL Macro Scenario

Methods	Disagreement p-value	Avg. Distance p-value
SMT25, IGA25, IAHP25	<2.2e-16	<2.2e-16
SMT50, IGA50, IAHP50	6.99e-03	1.28e-02
SMT100, IGA100, IAHP100	6.05e-01	7.59e-01

Table 13: Analysis of variance (ANOVA) comparing SMT, IGA and IAHP for MON Macro Scenario

Methods	Disagreement p-value	Avg. Distance p-value
SMT25, IGA25, IAHP25	<2.2e-16	<2.2e-16
SMT50, IGA50, IAHP50	9.092e-08	7.044e-08
SMT100, IGA100, IAHP100	3.78e-06	3.35e-06

Table 14: Analysis of variance (ANOVA) comparing SMT, IGA and IAHP for FALL Macro Scenario

Methods	**Disagreement p-value**	**Avg. Distance p-value**
SMT25, IGA25, IAHP25	3.14e-11	1.411e-11
SMT50, IGA50, IAHP50	1.41e-04	6.92e-03
SMT100, IGA100, IAHP100	1.28e-03	2.97e-03

Table 15: Analysis of variance (ANOVA) comparing SMT, IGA and IAHP for ESC Macro Scenario

6.3 Results for Research Question 2

RQ2 (Role of interaction) *Does SMT-based interactive prioritization produce improved prioritizations compared to non-interactive SMT-based prioritization?*

Our research hypothesis is that user knowledge plays an important role in requirement prioritization. RQ2 is designed to test such hypothesis. To assess the importance of the information elicited from the user, we compare the output of SMT before any pair is elicited from the user with the output of the same method after completing the elicitation of pairwise comparisons from the user to resolve ties.

6.3.1 Results for Research Question 2: ALL Macro Scenario

Macro scenario	Parameters	Value
ALL (49 req)	targetAlgorithm	SMT, non_I-SMT
	measurement	Disagreement, Average Distance
	maxElicitedPairs	0, 25, 50 & 100
	errorPerc	0% error rate

Table 16: Experimental settings for the RQ2: ALL Macro Scenario

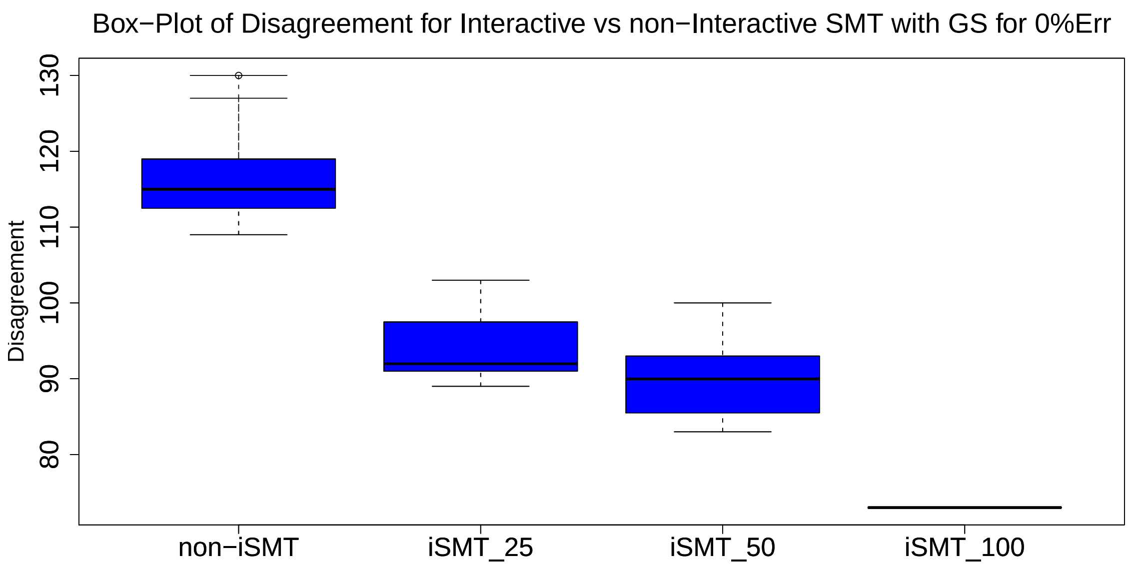

Figure 25: Box-Plot of Disagreement for Interactive vs non-Interactive SMT with GS for 0% user error, 49 Req

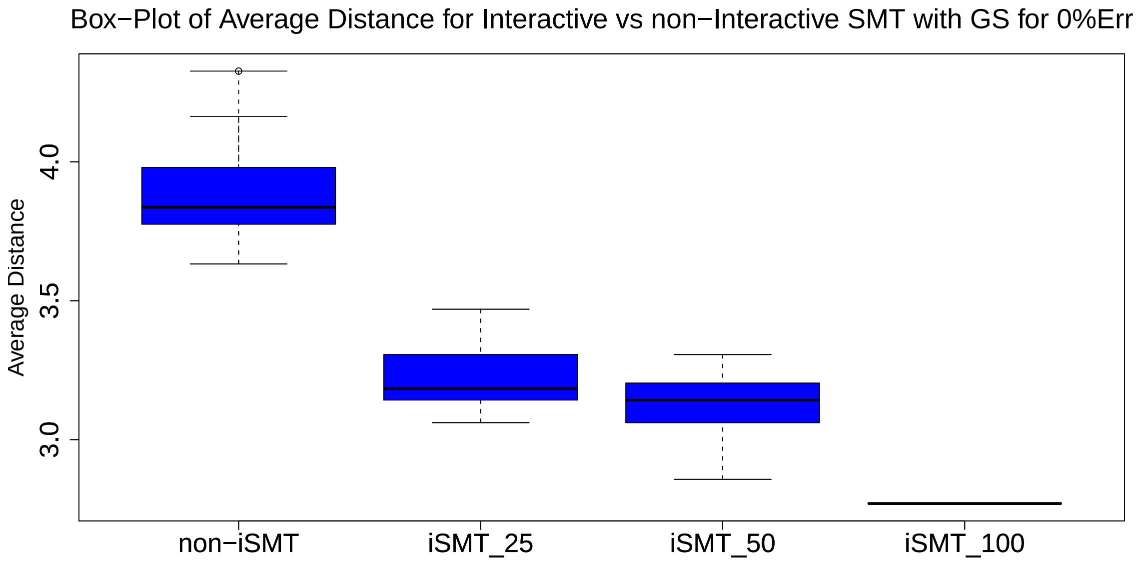

Figure 26: Box-Plot of Average Distance for Interactive vs non-Interactive SMT with GS for 0% user error, 49 Req

6.3.2 Results for Research Question 2: ESC Macro Scenario

Macro scenario	Parameters	Value
ESC (23 req)	targetAlgorithm measurement maxElicitedPairs errorPerc	SMT, non_I-SMT Disagreement, Average Distance 0, 25, 50 & 100 0% error rate

Table 17: Experimental settings for the RQ2: ESC Macro Scenario

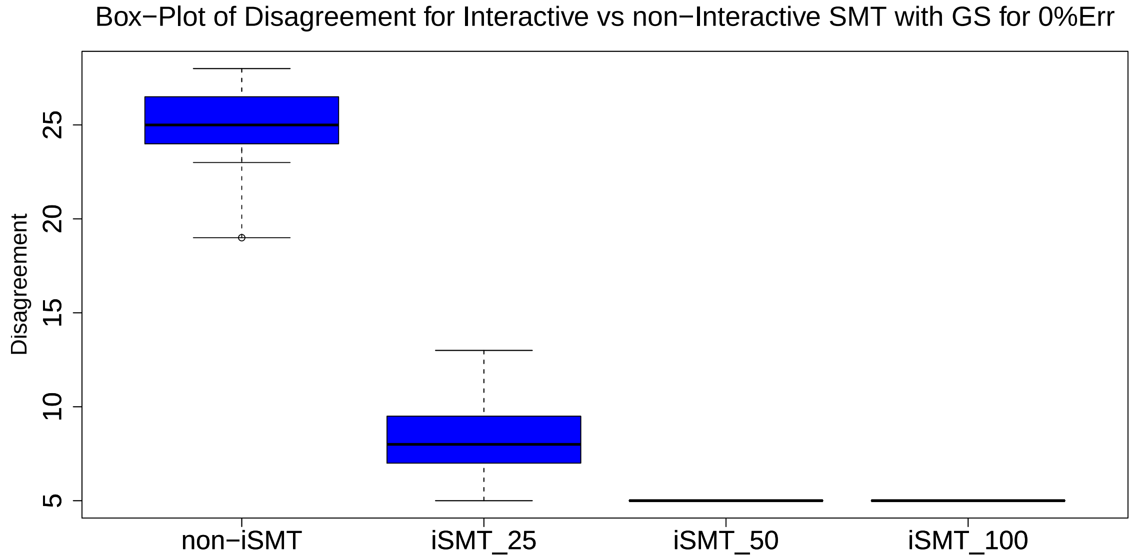

Figure 27: Box-Plot of Disagreement for Interactive vs non-Interactive SMT with GS for 0% user error, 23 Req

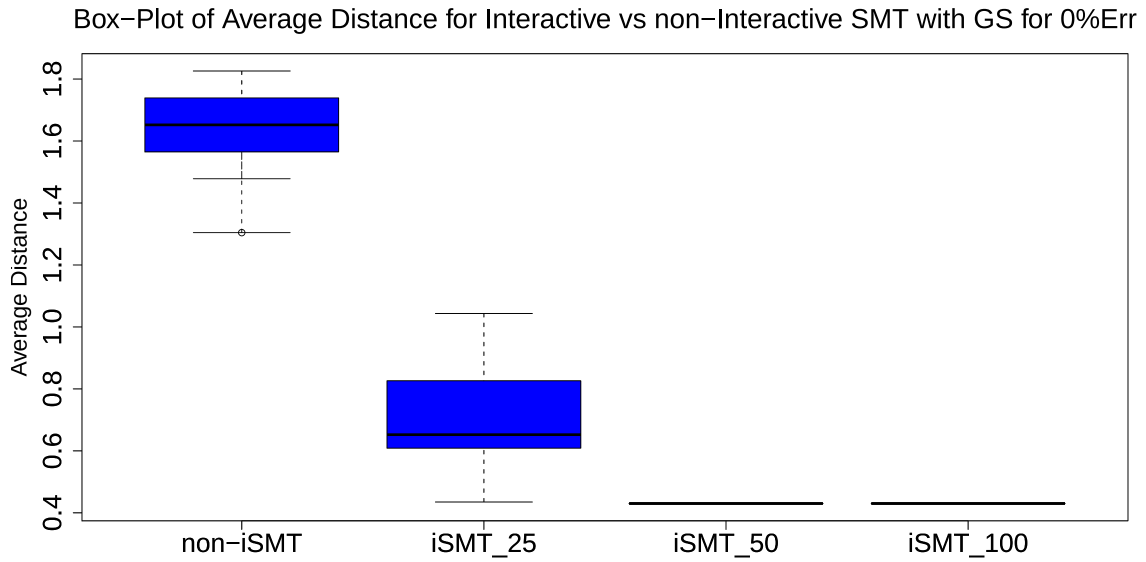

Figure 28: Box-Plot of Average Distance for Interactive vs non-Interactive SMT with GS for 0% user error, 23 Req

6.3.3 Results for Research Question 2: FALL Macro Scenario

Macro scenario	Parameters	Value
FALL (26 req)	targetAlgorithm	SMT, non_I-SMT
	measurement	Disagreement, Average Distance
	maxElicitedPairs	0, 25, 50 & 100
	errorPerc	0% error rate

Table 18: Experimental settings for the RQ2: FALL Macro Scenario

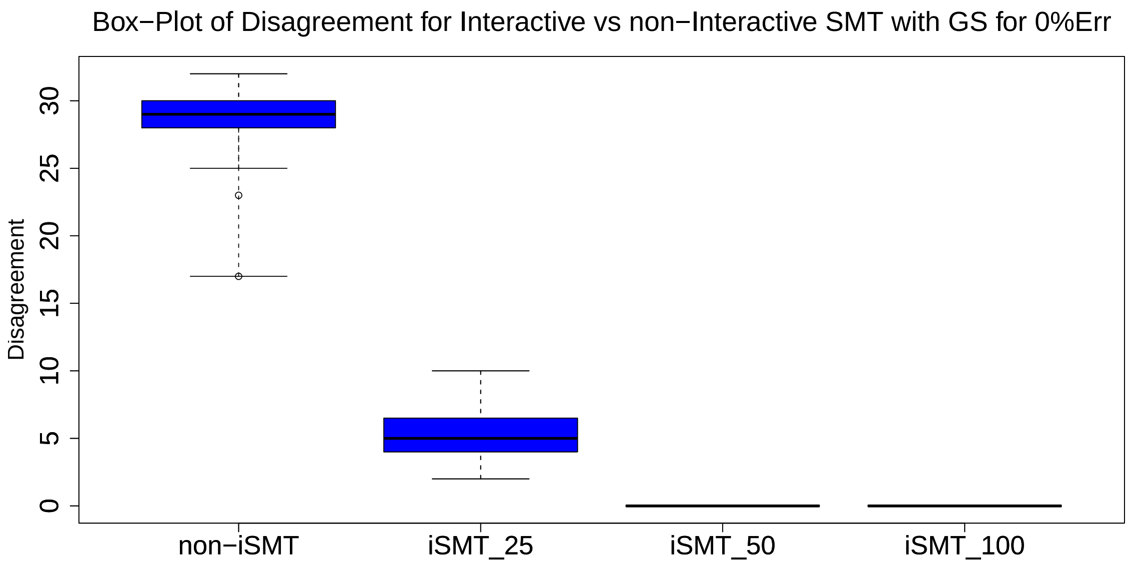

Figure 29: Box-Plot of Disagreement for Interactive vs non-Interactive SMT with GS for 0% user error, 26 Req

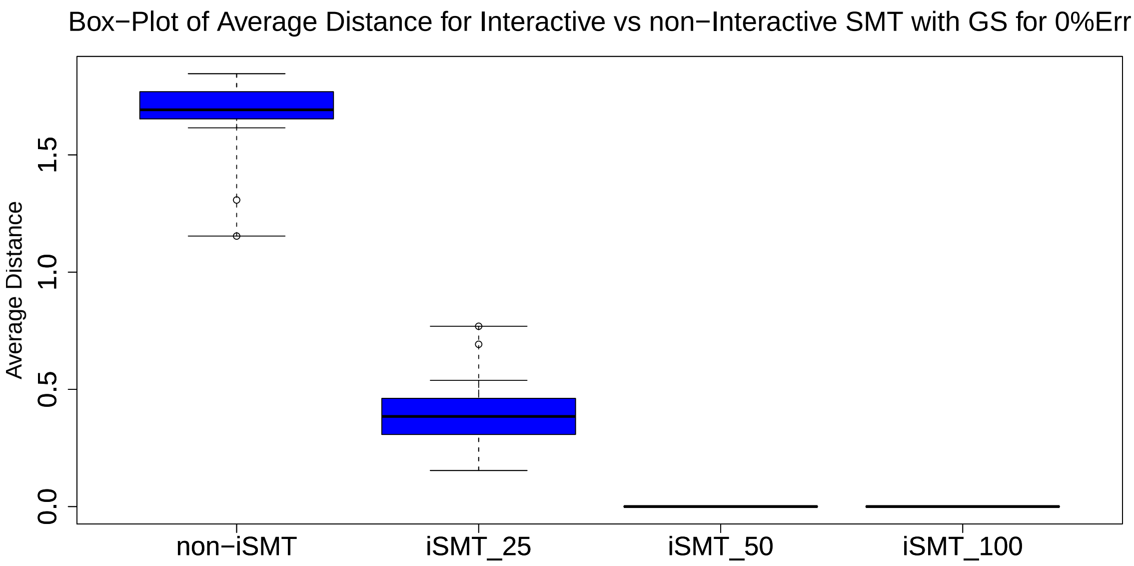

Figure 30: Box-Plot of Average Distance for Interactive vs non-Interactive SMT with GS for 0% user error, 26 Req

6.3.4 Results for Research Question 2: MON Macro Scenario

Macro scenario	Parameters	Value
MON (21 req)	targetAlgorithm measurement maxElicitedPairs errorPerc	SMT, non_I-SMT Disagreement, Average Distance 0, 25, 50 & 100 0% error rate

Table 19: Experimental settings for the RQ2: MON Macro Scenario

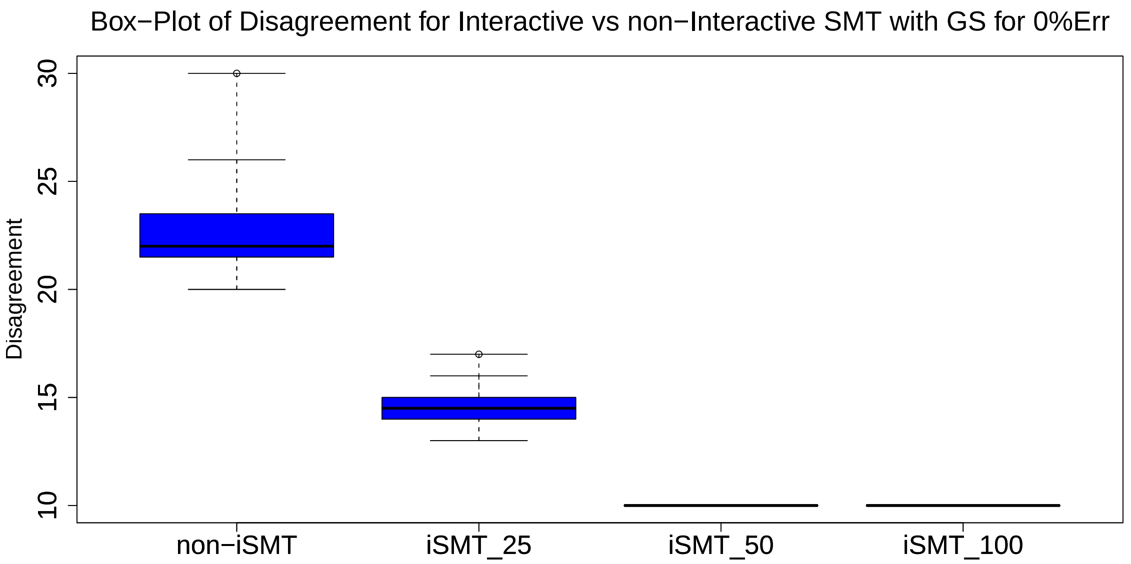

Figure 31: Box-Plot of Disagreement for Interactive vs non-Interactive SMT with GS for 0% user error, 21 Req

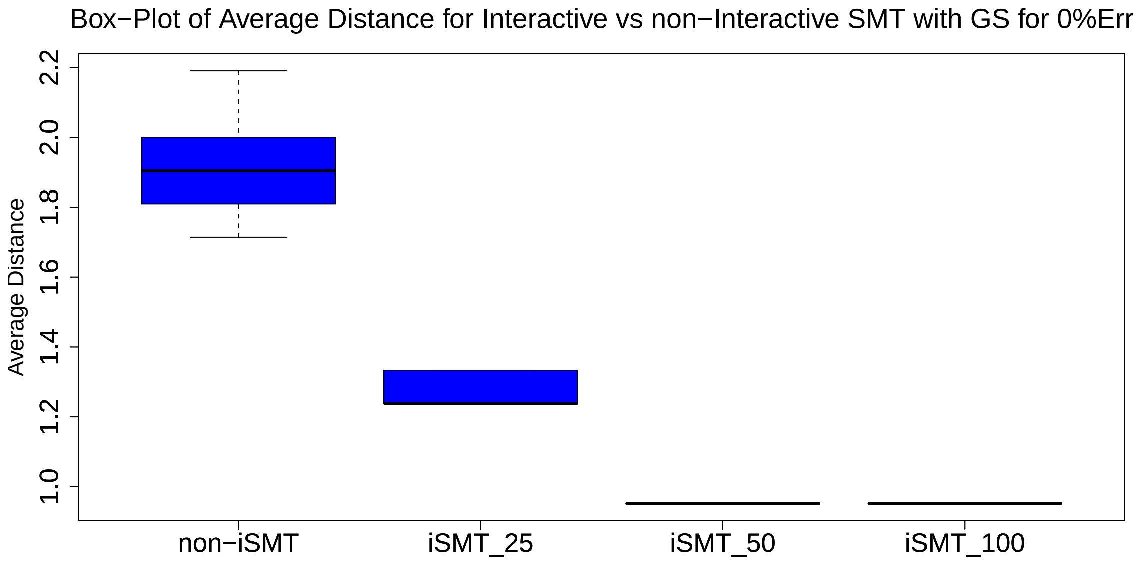

Figure 32: Box-Plot of Average Distance for Interactive vs non-Interactive SMT with GS for 0% user error, 21 Req

6.4 Results for Research Question 3

RQ3 (Robustness) *Is the SMT-based method more robust than IAHP and IGA with respect to errors committed by the user during the elicitation of pairwise comparisons?*

In order to test the robustness of SMT in comparison with IAHP and IGA at increasing user error rates, we simulate such errors by means of a simple stochastic model. We fix a probability of committing an elicitation error, say p_e. Then, during the execution of the SMT, IAHP and IGA algorithms, whenever a pairwise comparison is elicited from the user, we generate a response in agreement with the GS with probability $1 - p_e$ and we generate an error (i.e., a response in disagreement with the GS) with probability p_e. We varied the probability of user error p_e from 5% to 20%.

6.4.1 Results for Research Question 3: ALL Macro Scenario

Macro scenario	Parameters	Value
ALL (49 req)	targetAlgorithm	SMT, IGA, IAHP
	measurement	Disagreement, Average Distance
	maxElicitedPairs	50, 100
	errorPerc	0%, 5%, 10%, 20%

Table 20: Experimental settings for the RQ3: ALL Macro Scenario

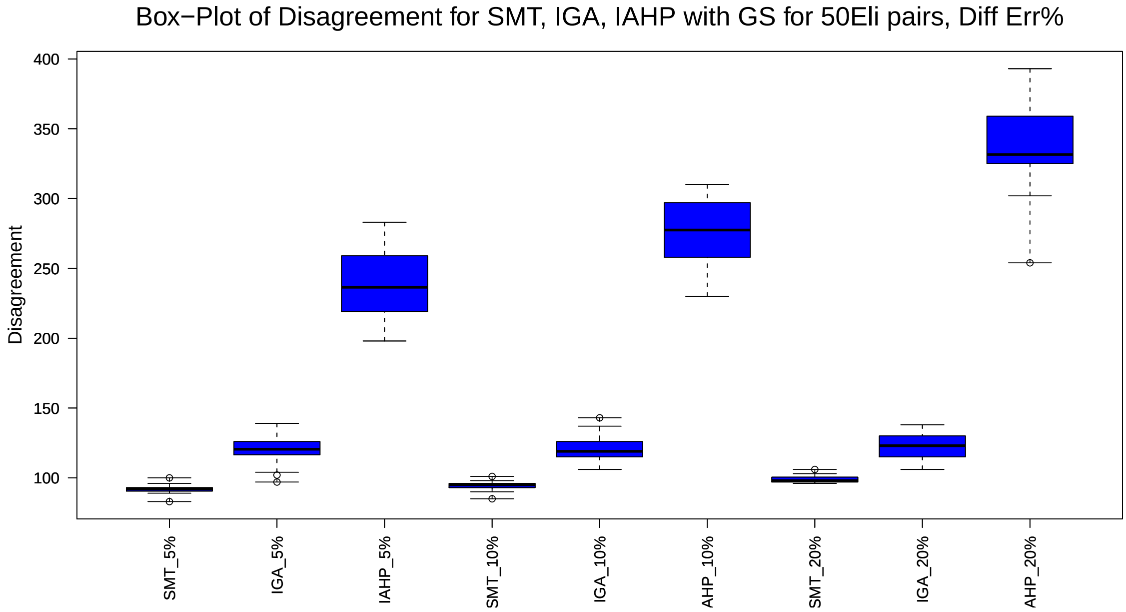

Figure 33: Box-Plot of Disagreement for SMT, IGA and IAHP with GS for 50Eli pairs, different user error%, 49 Req

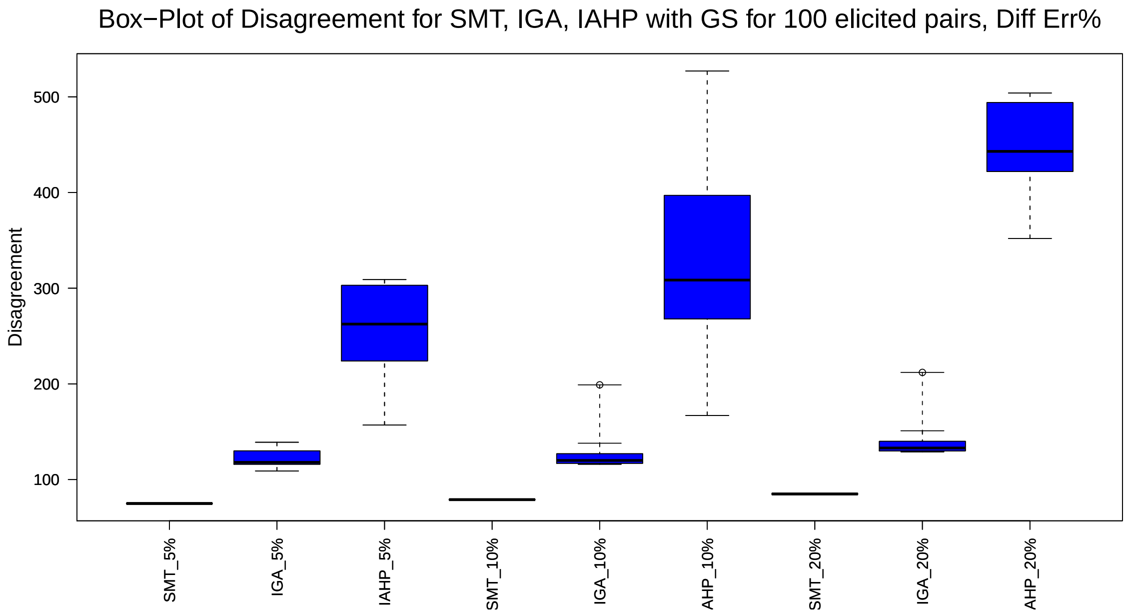

Figure 34: Box-Plot of Disagreement for SMT, IGA and IAHP with GS for 100 elicited pairs, different user error%, 49 Req

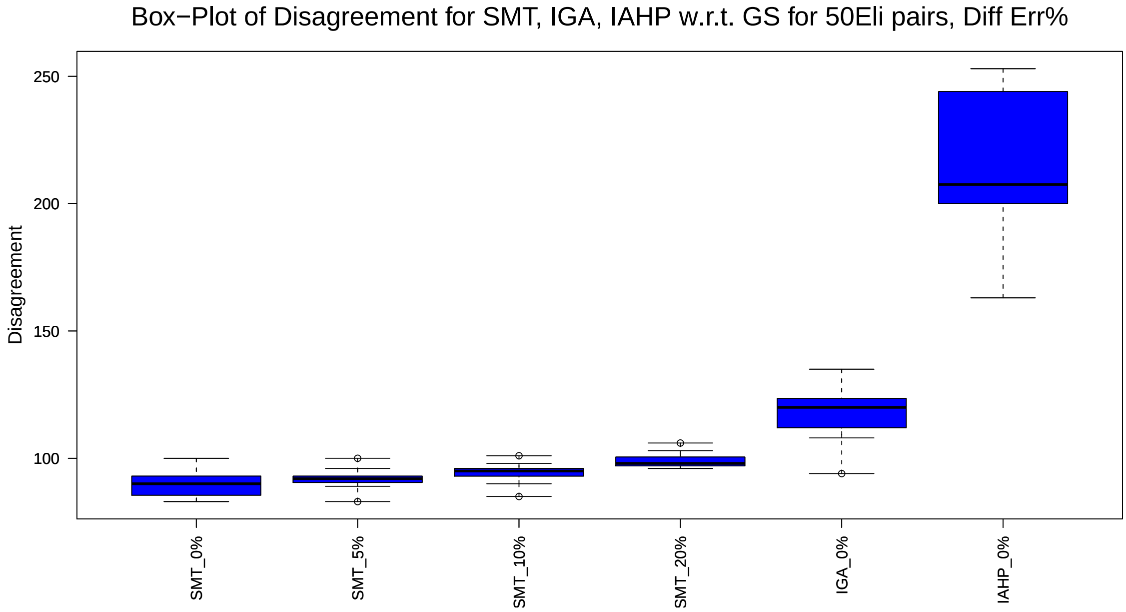

Figure 35: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 50Eli pairs, different user error%, 49 Req

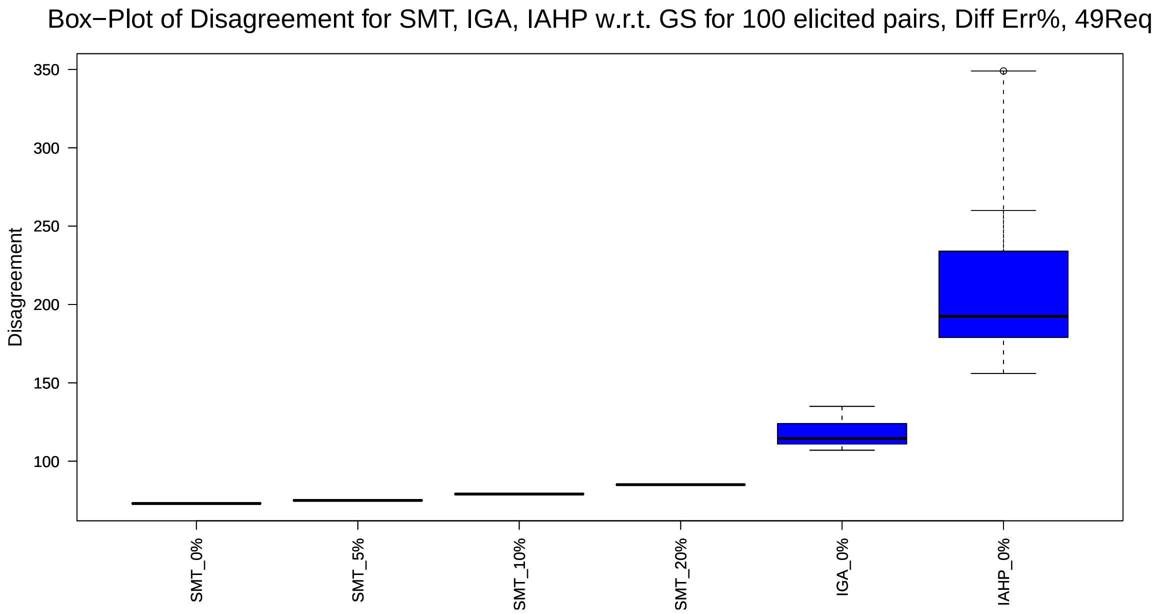

Figure 36: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 100 elicited pairs, different user error%, 49 Req

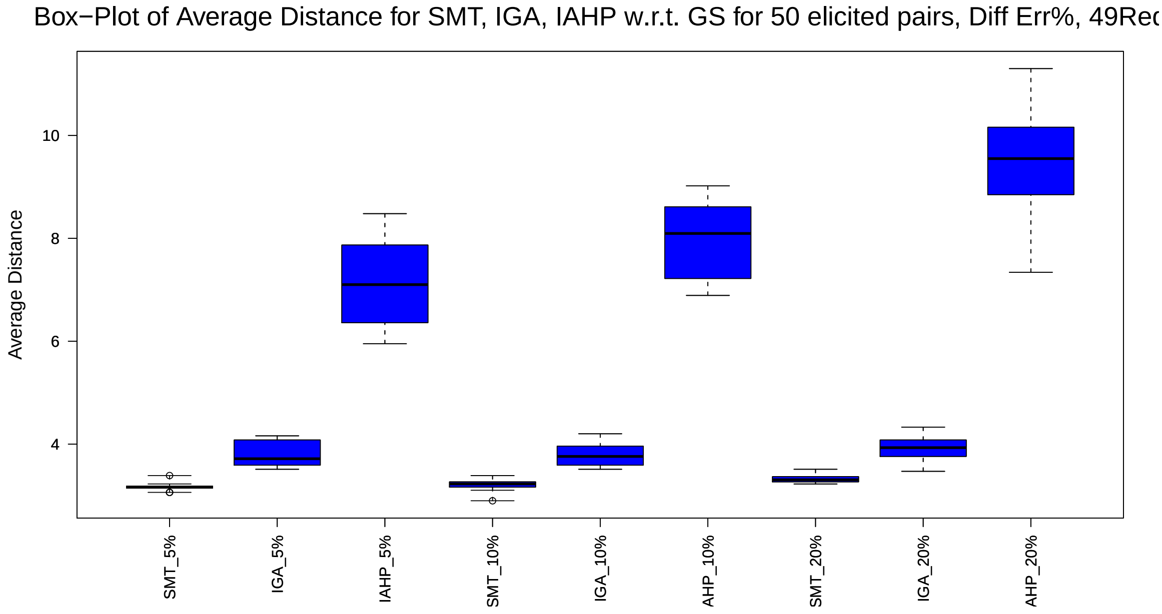

Figure 37: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 50 elicited pairs, different user error%, 49 Req

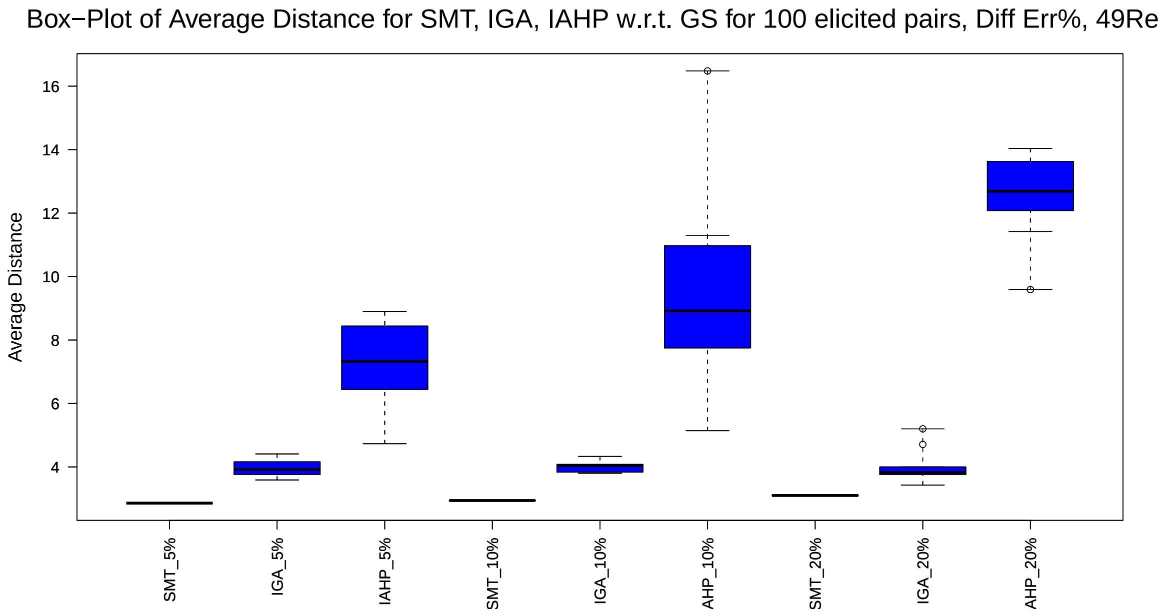

Figure 38: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 100 elicited pairs, different user error%, 49 Req

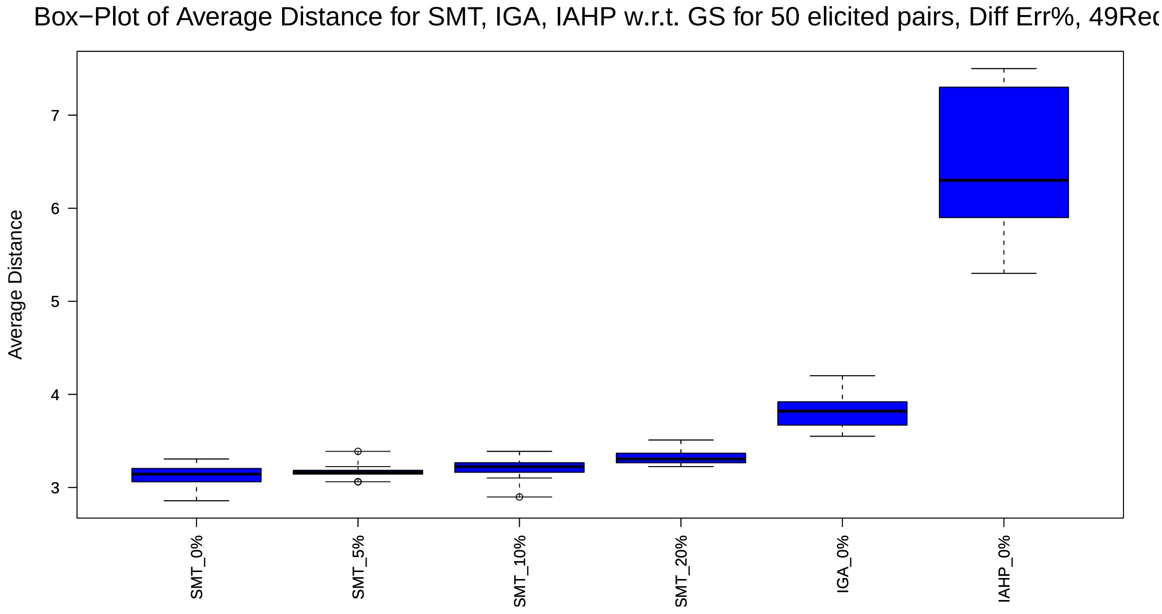

Figure 39: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 50 elicited pairs, different user error%, 49 Req

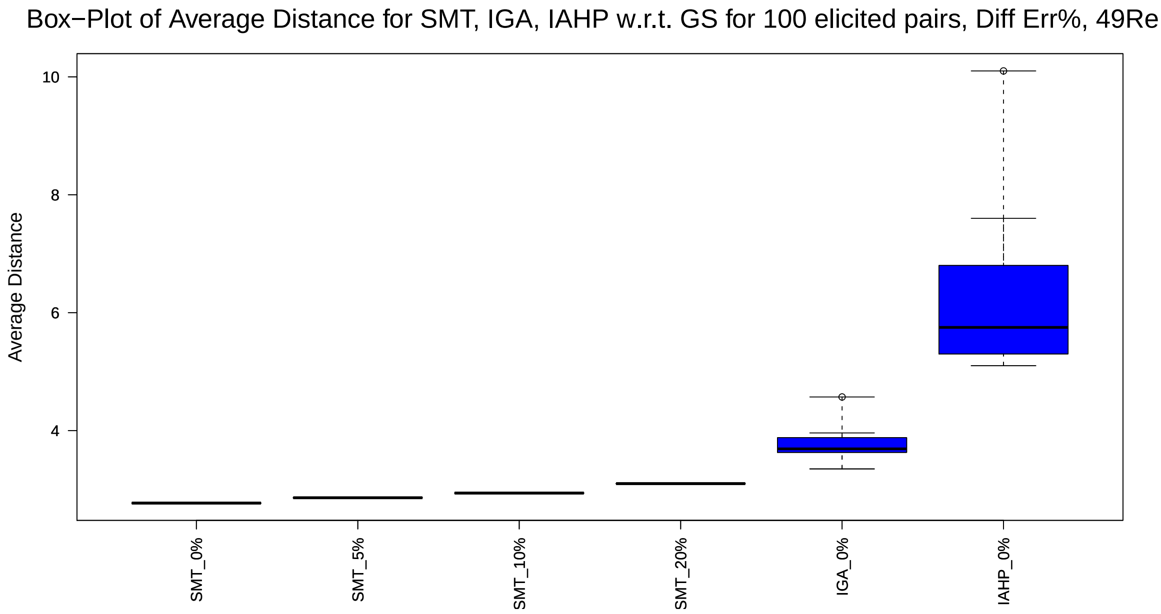

Figure 40: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 100 elicited pairs, different user error%, 49 Req

6.4.2 Results for Research Question 3: ESC Macro Scenario

Macro scenario	Parameters	Value
ESC (23 req)	targetAlgorithm	SMT, IGA, IAHP
	measurement	Disagreement, Average Distance
	maxElicitedPairs	50, 100
	errorPerc	0%, 5%, 10%, 20%

Table 21: Experimental settings for the RQ3: ESC Macro Scenario

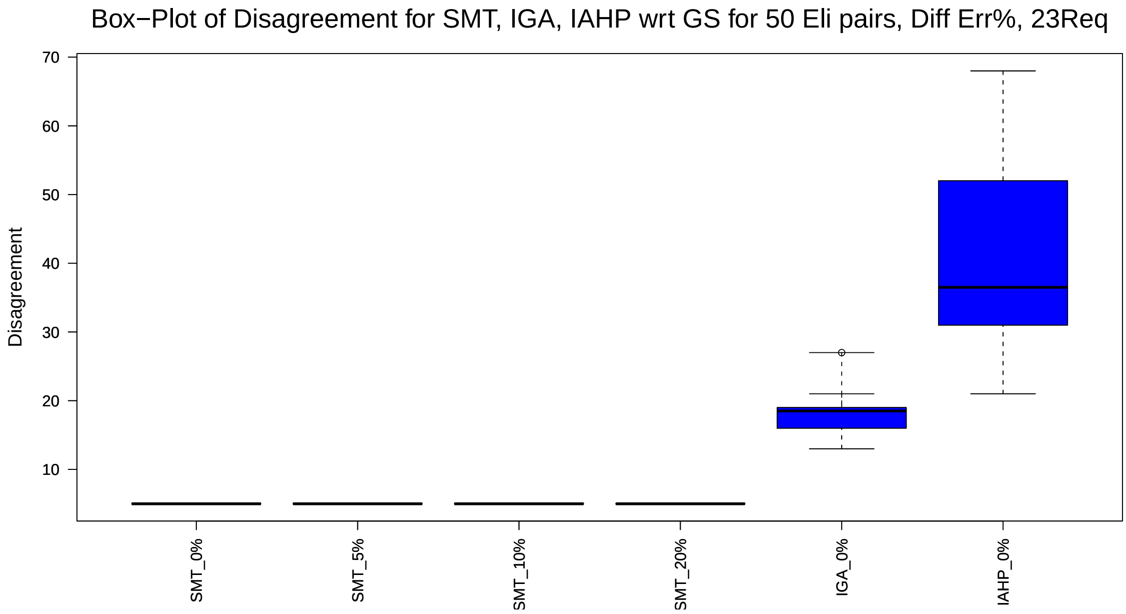

Figure 41: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, different user error%, 23 Req

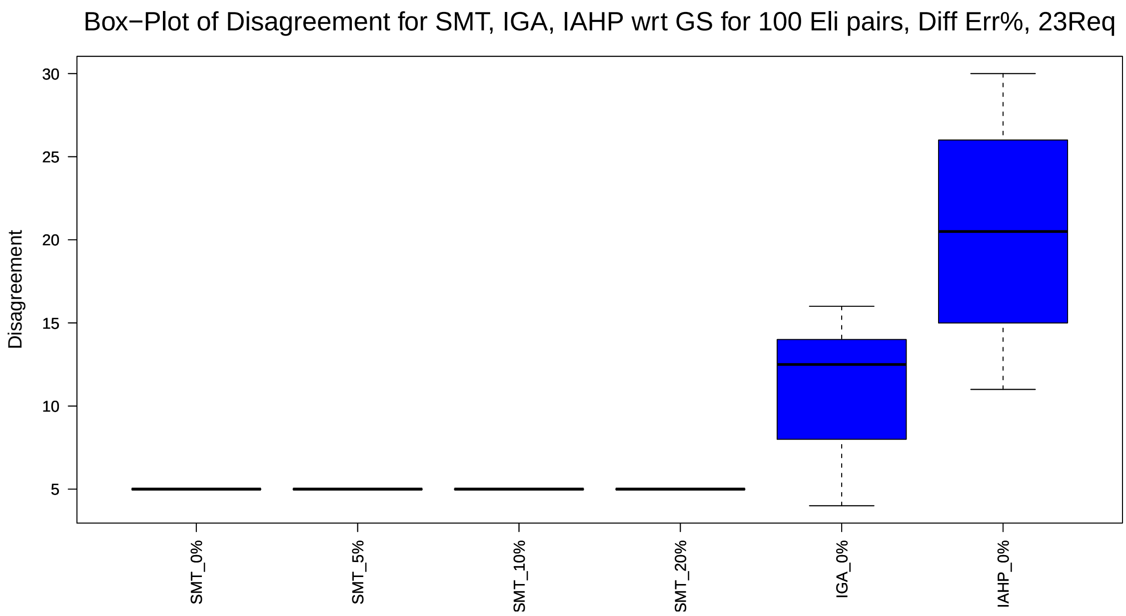

Figure 42: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, different user error%, 23 Req

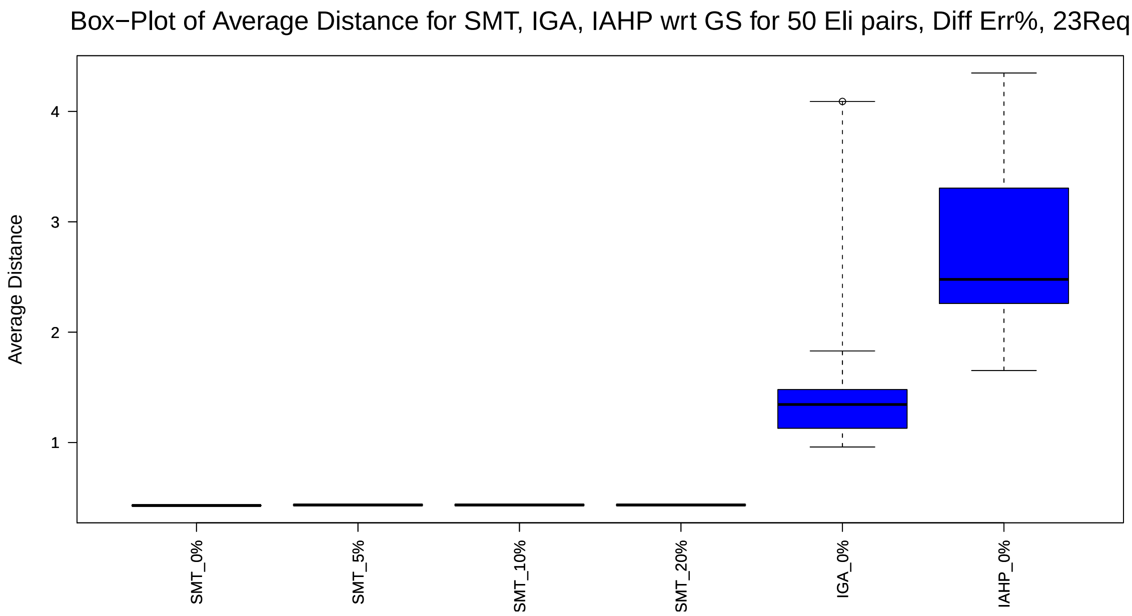

Figure 43: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, different user error%, 23 Req

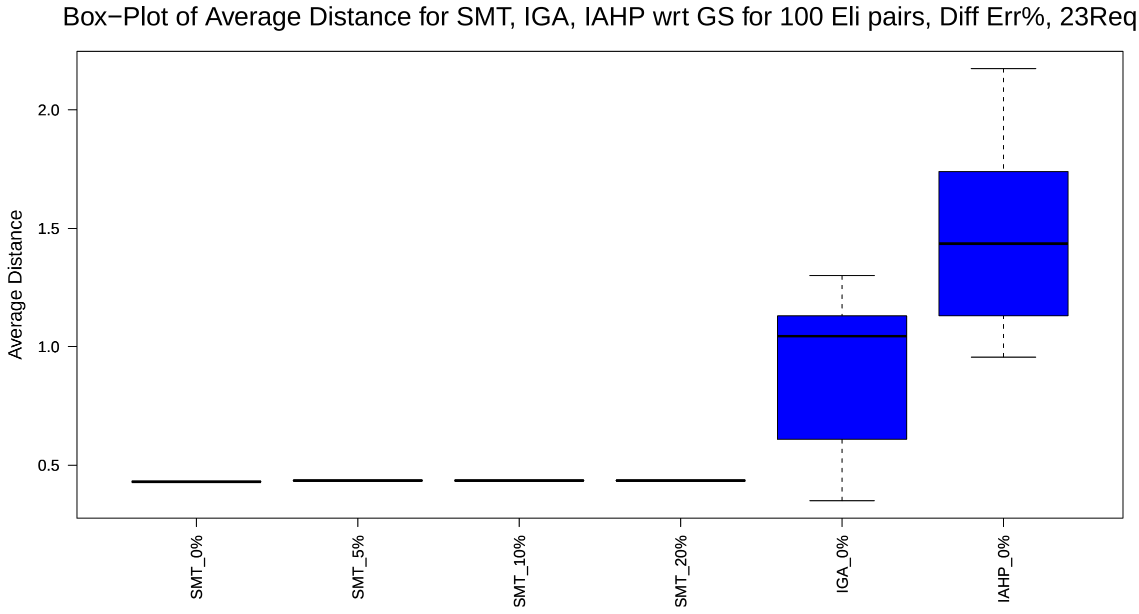

Figure 44: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, different user error%, 23 Req

6.4.3 Results for Research Question 3: FALL Macro Scenario

Macro scenario	Parameters	Value
FALL (26 req)	targetAlgorithm measurement maxElicitedPairs errorPerc	SMT, IGA, IAHP Disagreement, Average Distance 50, 100 0%, 5%, 10%, 20%

Table 22: Experimental settings for the RQ3: FALL Macro Scenario

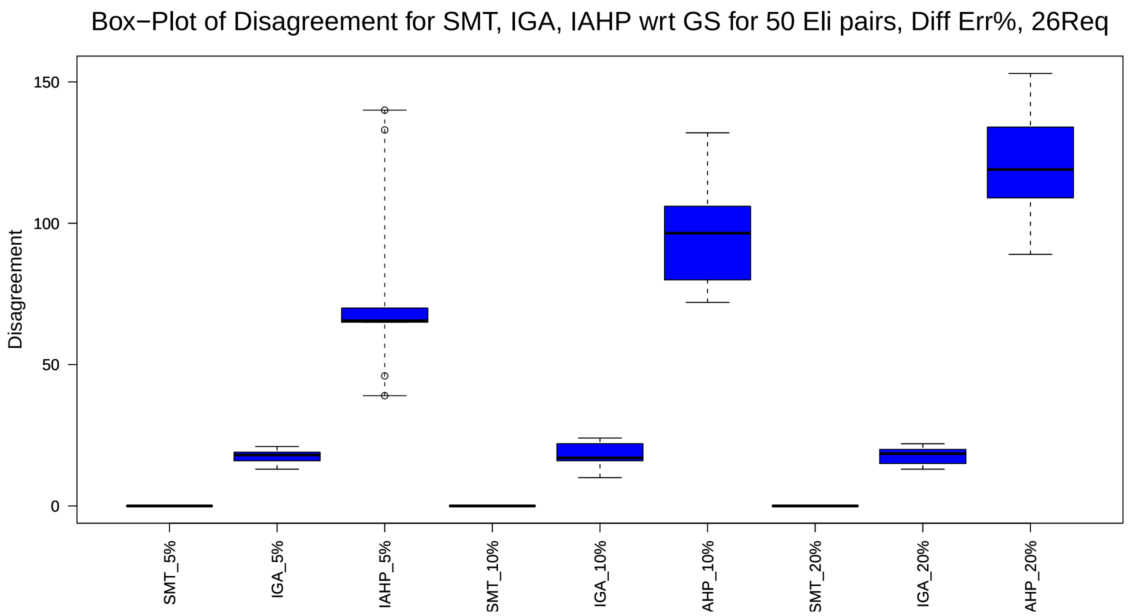

Figure 45: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, different user error%, 26 Req

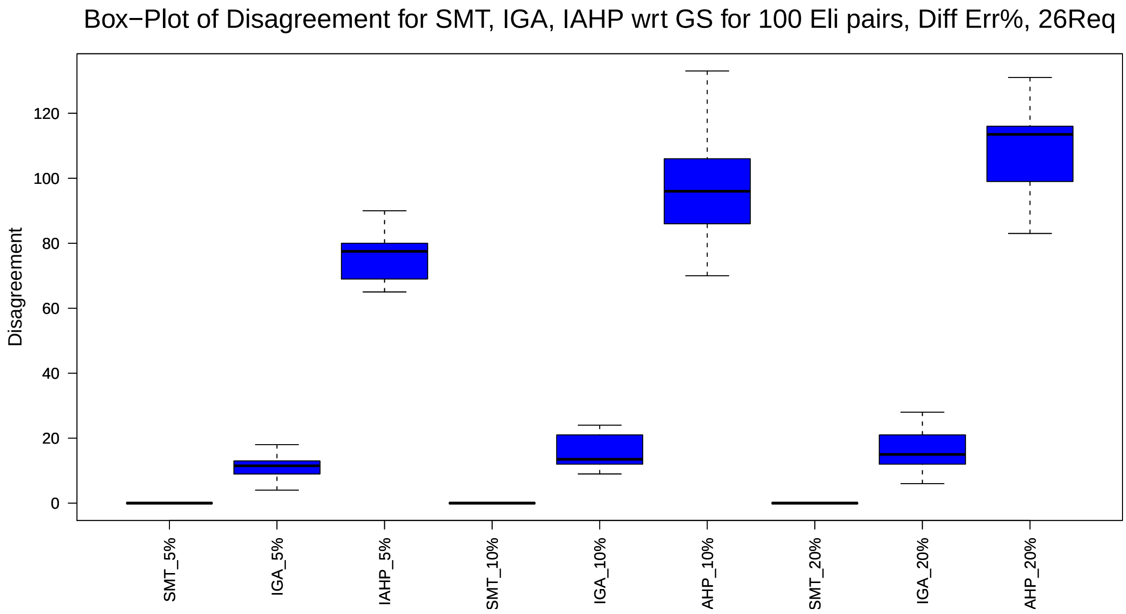

Figure 46: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, different user error%, 26 Req

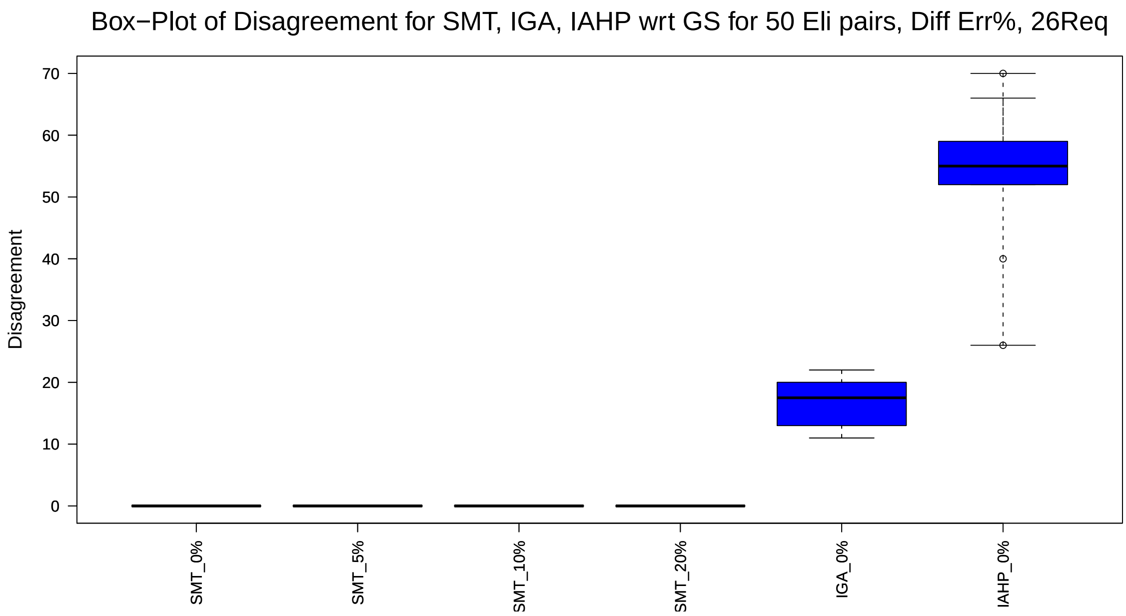

Figure 47: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, different user error%, 26 Req

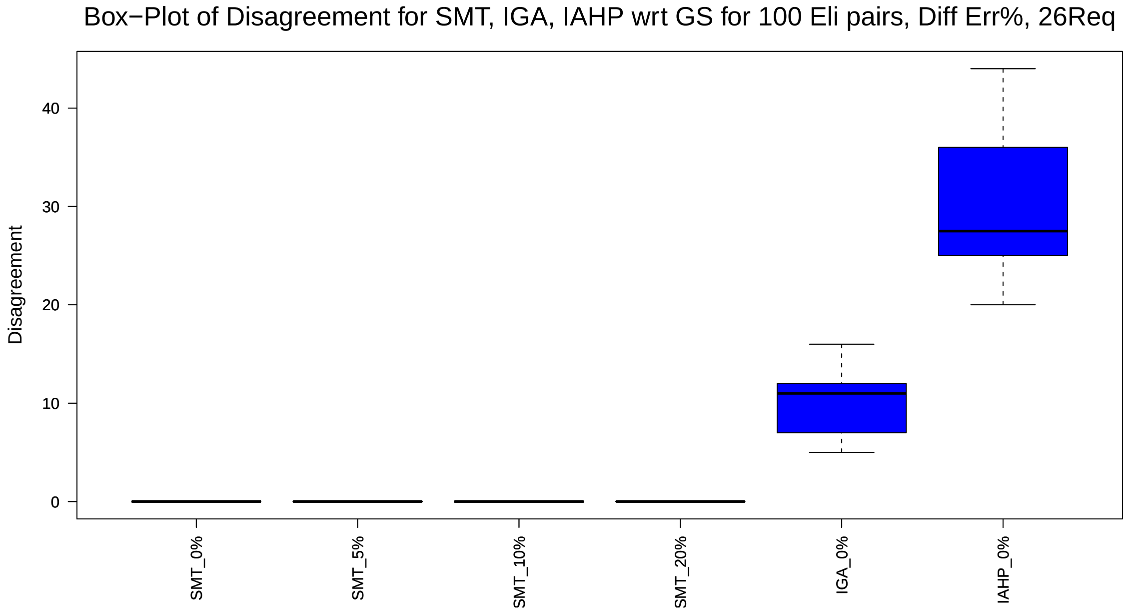

Figure 48: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, different user error%, 26 Req

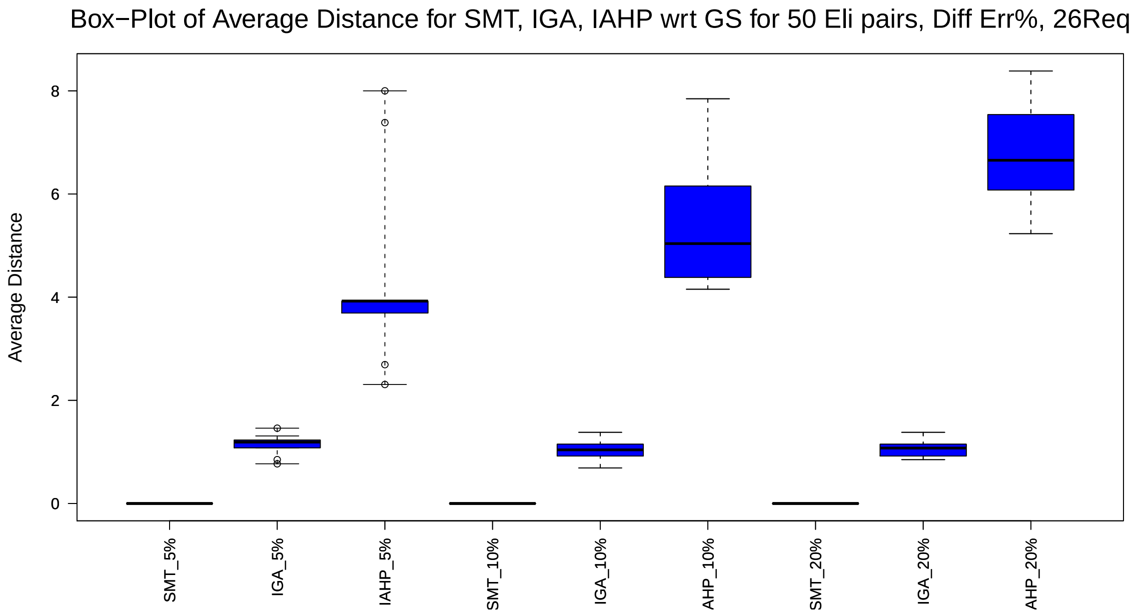

Figure 49: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, different user error%, 26 Req

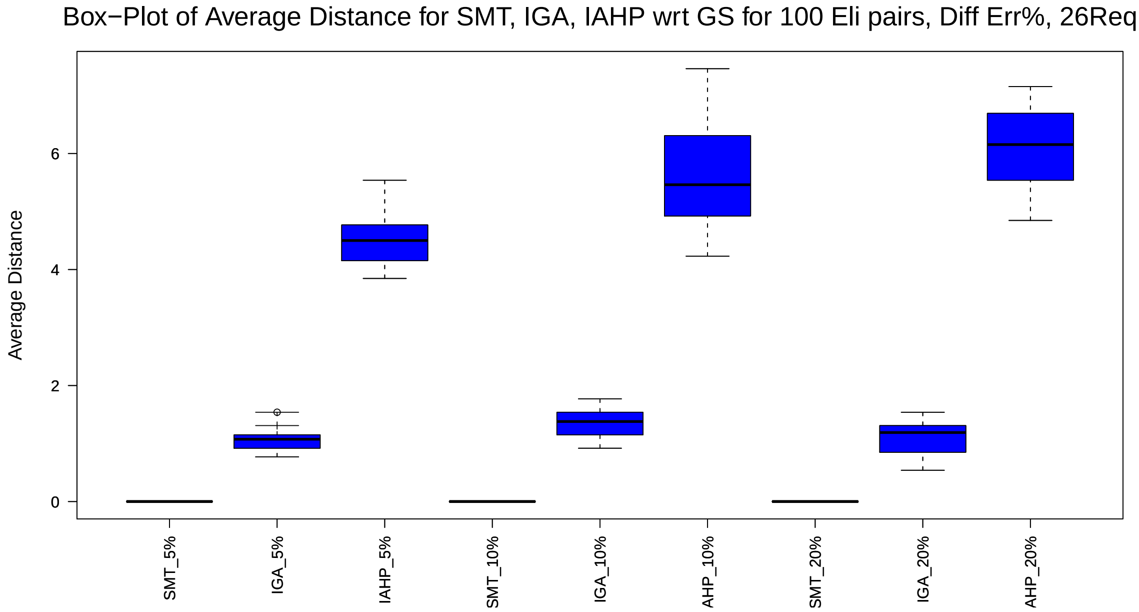

Figure 50: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, different user error%, 26 Req

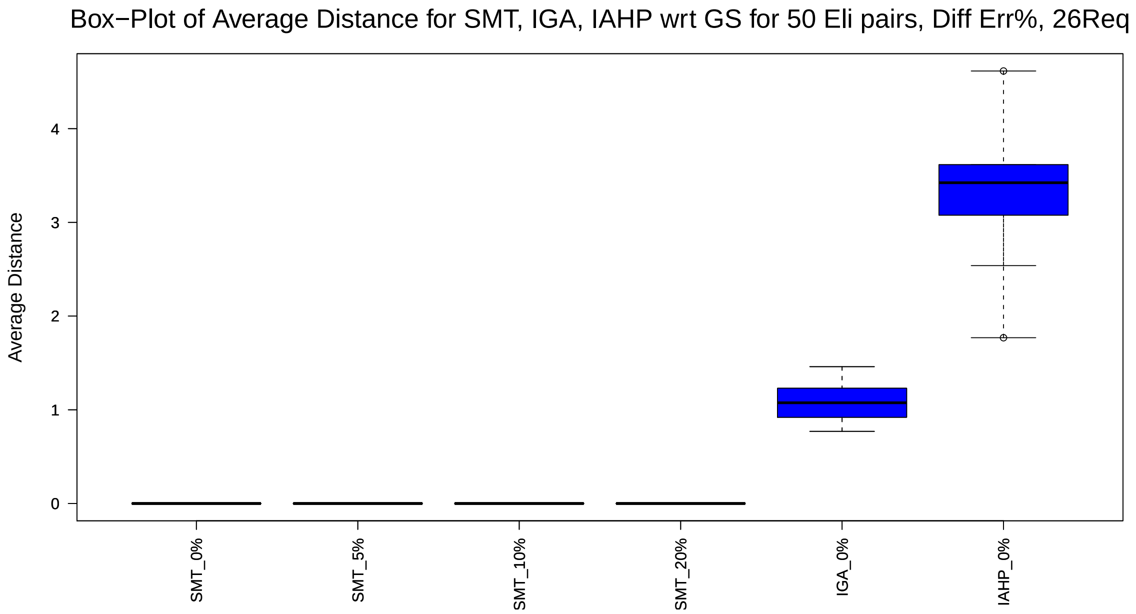

Figure 51: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, different user error%, 26 Req

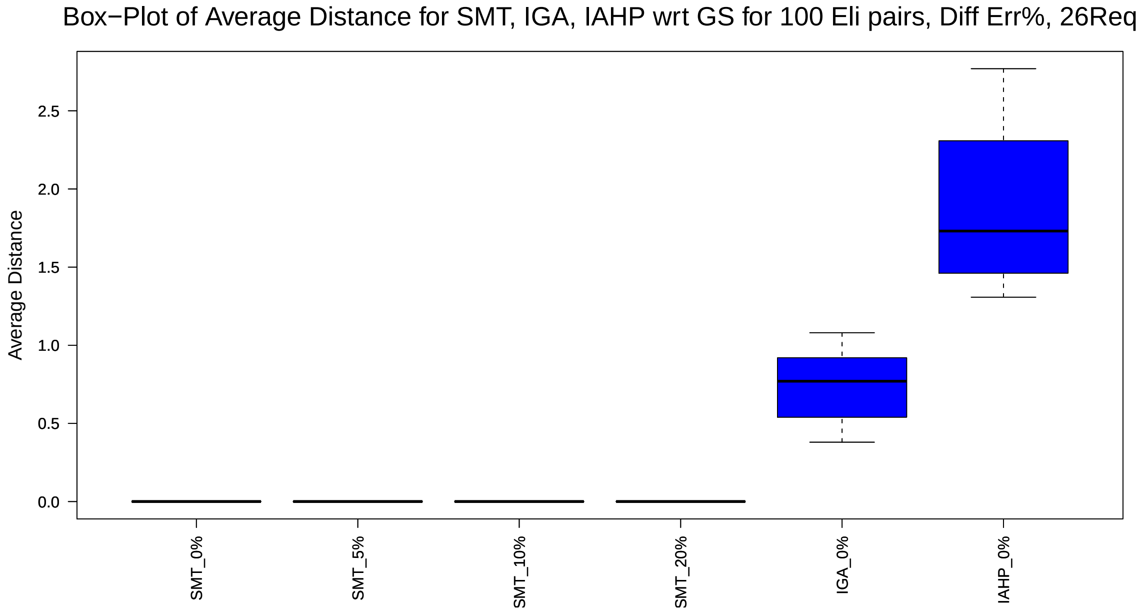

Figure 52: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, different user error%, 26 Req

6.4.4 Results for Research Question 3: MON Macro Scenario

Macro scenario	Parameters	Value
MON (21 req)	targetAlgorithm measurement maxElicitedPairs errorPerc	SMT, IGA, IAHP Disagreement, Average Distance 50, 100 0%, 5%, 10%, 20%

Table 23: Experimental settings for the RQ3: MON Macro Scenario

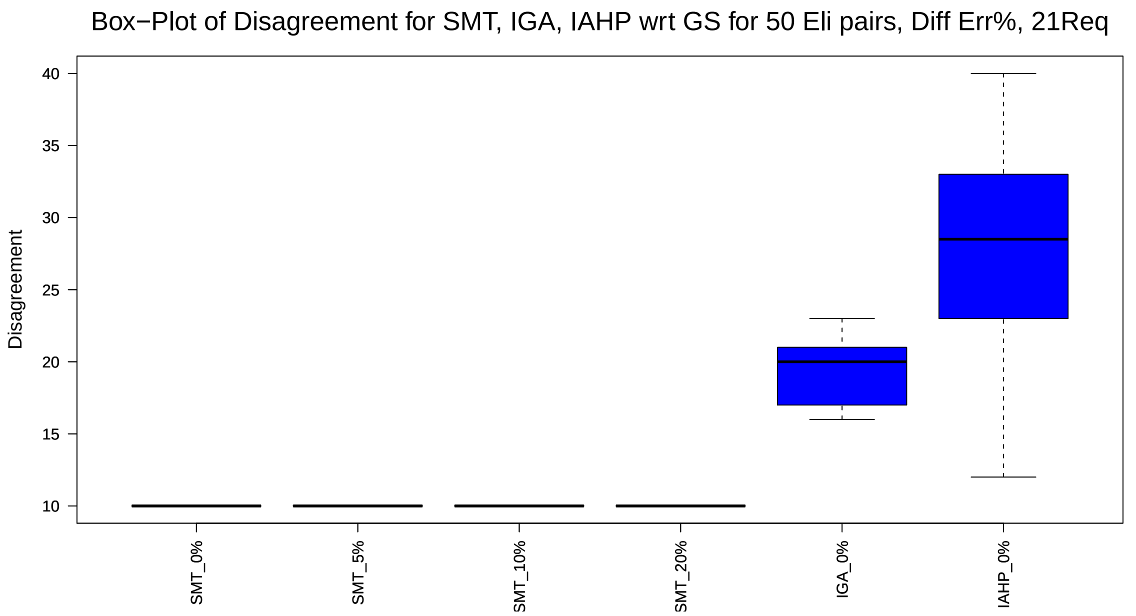

Figure 53: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, different user error%, 21 Req

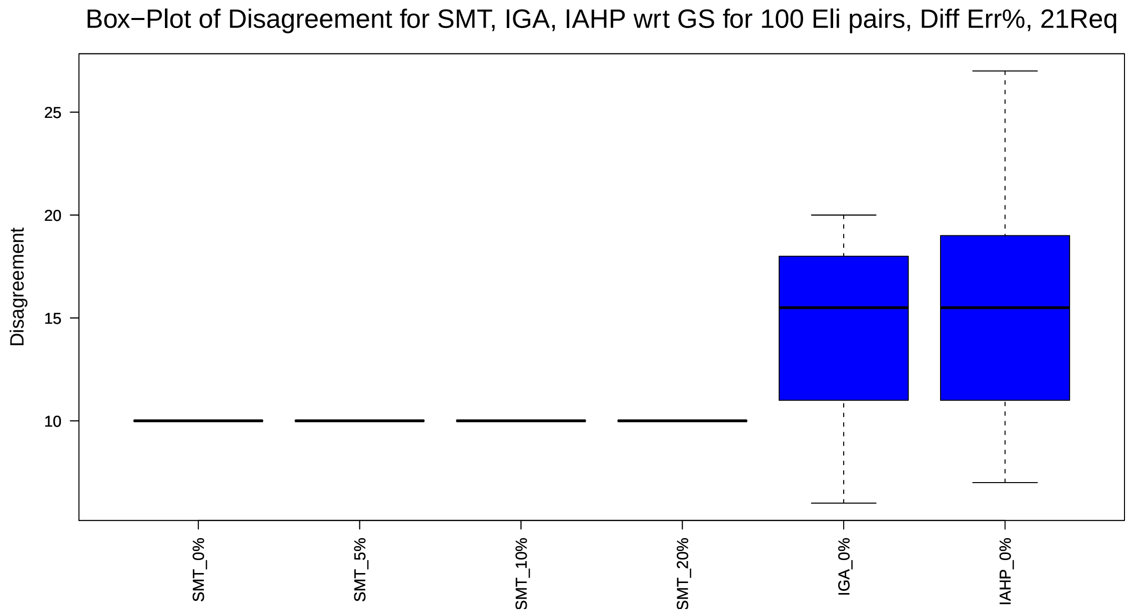

Figure 54: Box-Plot of Disagreement for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, different user error%, 21 Req

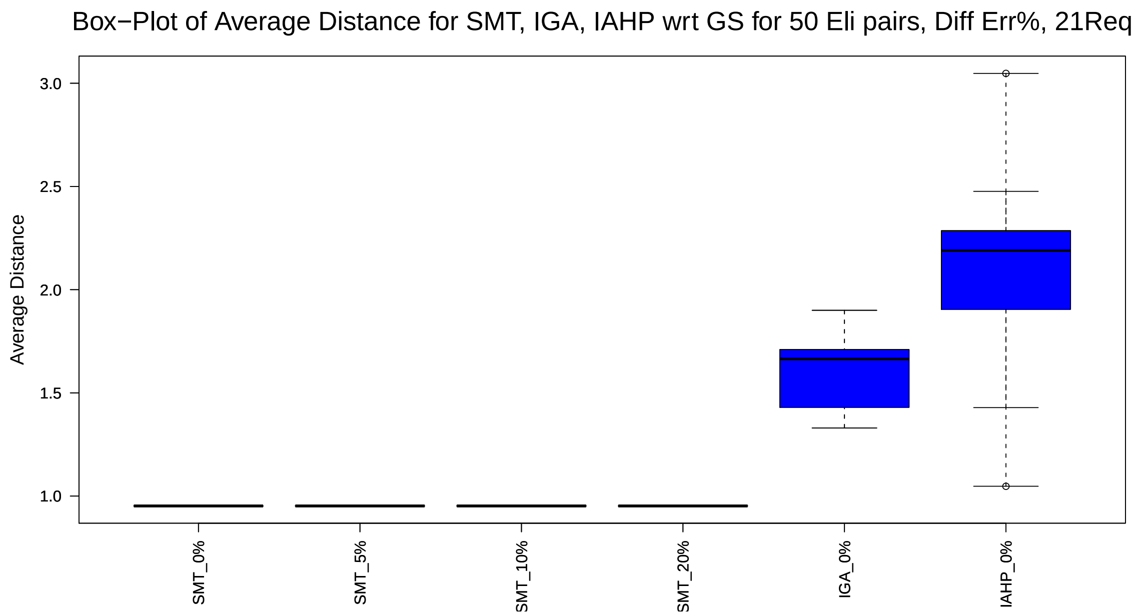

Figure 55: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 50 Eli pairs, different user error%, 21 Req

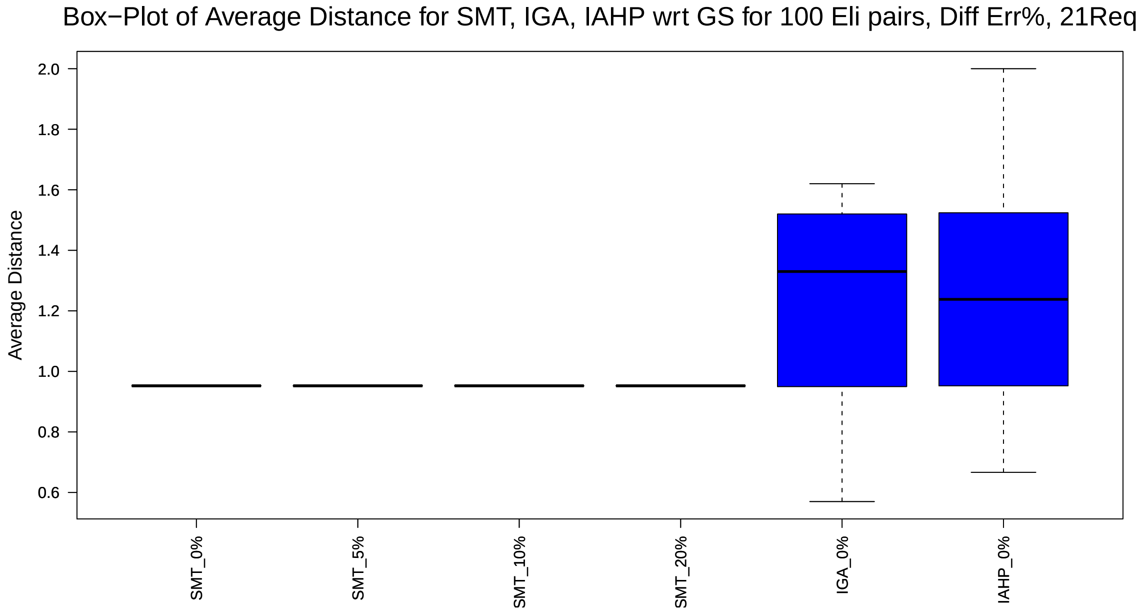

Figure 56: Box-Plot of Average Distance for SMT, IGA and IAHP w.r.t. GS for 100 Eli pairs, different user error%, 21 Req

6.5 Results for Research Question 4

RQ4 (Performance) *What is the machine time necessary to execute the constraint solver during the prioritization process, compared to the execution time of the optimization algorithms involved in IAHP and IGA?*

This research question deals with the time performance of the three methods being compared: SMT, IAHP and IGA. Low execution time of the constraint solving/optimization step is fundamental to allow online usage of the proposed method. In fact, the time between an elicitation and the next one is determined by the execution time of the algorithm that performs the required constraint solving/optimization steps and then determines which pairwise comparisons are to be elicited next. Such time should be acceptable from the point of view of the decision maker who answers the pairwise comparison questions.

Pairs	SMT	IGA	IAHP
25	46s	640s	4s
50	51s	840s	6s
100	50s	1080s	128s

Table 24: Prioritization time (system time) required by SMT, IGA and IAHP for the ALL scenario

Table 24 shows the comparison among execution times of SMT, IGA and IAHP. All execution times in the table are obtained from the system time of each execution and are measured in seconds. The experiments were performed on an Intel Core2 Duo 2.10GHz machine with 3GB of memory, running the 32bit Windows Vista OS. The overall prioritization process time is much lower when using SMT as compared to IGA. IAHP is more efficient than SMT at 25 and 50 elicited pairs, but its performance seems to suffer a non-linear degradation when the number of elicitations increases. In fact, at 100 elicited pairs, the execution time of IAHP jumps from 4s to 128s. On the contrary, SMT seems to have little dependence on the number of elicitations. At 100 elicited pairs it remains almost constant and it is substantially inferior than IAHP.

7 Discussion

Based on the data collected from our experiments, we answered positively to all our research questions. When comparing the prioritizations produced by the considered algorithms with GS, both in terms of disagreement and of position distance, interactive SMT outperforms substantially IGA and IAHP (RQ1). Moreover, interactive SMT outperforms non-interactive SMT (RQ2), hence showing that user input plays a key role in producing a prioritization of requirements which gets close to the reference, *Gold Standard* one. The behavior of the algorithm is robust with respect to the presence of elicitation errors committed by the user (RQ3). Up to 20% user error rate, SMT outperforms IGA and IAHP, even when these are run at 0% user error rate. Finally, our proposed SMT-based interactive approach consumes much less system-time compared to IGA. This is true in comparison to IAHP only at a high number of elicited pairs (e.g., 100). The estimated time between consecutive pairwise elicitations (0.001s–0.95s) is compatible with online usage of the tool.
If we look at the average distance of each requirement from the position it has in the GS, we can see that such a distance is quite low (2–3 positions). This indicates that the approximate solution produced by our method is close enough to the ideal solution to be usable in practice. Of course, further studies are needed to evaluate the acceptability of the approximate solution by the users.

8 Conclusions

In this report we presented all the results of SMT-based approach to the requirements prioritization problem relying on Satisfiability Modulo Theory techniques. The approach is interactive and is based on a pairwise preference elicitation process to extract relevant knowledge from the decision makers that is encoded as constraints. This knowledge is then combined with the knowledge related to the ordering criteria induced by requirements characteristics, such as implementation costs, values for the users, implementation dependencies, that are also encoded as constraints.

The aim is that of minimizing the user decision-making effort, in terms of number of pairwise comparison to be specified, increasing as much as possible the accuracy of the final requirements ranking.

To validate the approach we conducted experiments on a set of requirements from a real health-care project (ACube). Results show that our approach overcomes other state-of-the-art interactive approaches such as IAHP and IGA, both in terms of disagreement between the ideal solution and the solution obtained via the methods and in terms of execution time. Moreover, the importance of the interaction has been verified with the decision maker, which increases substantially the performance of the SMT approach with respect to a non interactive version of the same algorithm. The method is also robust with respect to user errors, maintaining good performance even when the percentage of errors is around 20%.

References

[1] Hoh Peter In, David Olson, and Tom Rodgers. Multi-criteria preference analysis for systematic requirements negotiation. In *COMPSAC 2002*, pages 887–892, 2002.

[2] Joachim Karlsson. Software Requirements Prioritizing. In *Proceedings of 2nd International Conference on Requirements Engineering (ICRE '96)* , pages 110–116, April 1996.

[3] J. Karlsson and K. Ryan. A cost-value approach for prioritizing requirements. *IEEE Software*, 14(5):67–74, 1997.

[4] Soren Lauesen. *Software requirements: styles and techniques*. Addison Wesley, 2002.

[5] F. Moisiadis. Prioritising software requirements. In *SERP 2002*, June 2002.

[6] S. Sivzittian and B. Nuseibeh. Linking the Selection of Requirements to Market Value: A Portfolio - Based Approach. In *REFSQ 2001*, 2001.

[7] Karl E. Wiegers. *Software Requirements. Best Practices*. Microsoft Press, 1999.

[8] Thomas L. Saaty and Luis G. Vargas. *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*. Kluwer Academic, 2000.

[9] Paolo Avesani, Cinzia Bazzanella, Anna Perini, and Angelo Susi. Facing scalability issues in requirements prioritization with machine learning techniques. In *RE 2005*, pages 297–306, 2005.

[10] P. T. Harker. Incomplete Pairwise Comparisons in the Analytic Hierarchy Process. *Math. Modelling*, 9(11):837 – 848, 1987.

[11] P. Tonella, A. Susi, and F. Palma. Using Interactive GA for Requirements Prioritization. In *2nd International Symposium on Search Based Software Engineering*, pages 57–66. IEEE, 2010.

[12] Renzo Andrich, Francesco Botto, Valerio Gower, Chiara Leonardi, Oscar Mayora, Lucia Pigini, Valentina Revolti, Luca Sabatucci, Angelo Susi, and Massimo Zancanaro. ACube: User-Centred and Goal-Oriented techniques. Technical report, Fondazione Bruno Kessler - IRST, 2010.