Complete Instantiation Strategy for Approximate Solution of TCSP

Priti Chandra, Arun K. Pujari

A I Lab, University of Hyderabad, India
priti_murali@yahoo.com

Abstract. Interval algebra(IA) based temporal constraint satisfaction problems (TCSPs) are useful in formulating diverse problems. The usual approach to solve IA networks is based on partial instantiation strategy - backtrack search for exact solution. To the best of our knowledge, determining approximate solution for TCSPs is not addressed so far. In this paper we propose a new complete instantiation strategy based on a complete algorithm to determine an approximate solution of IA networks. We identify a property of constraints called *nastiness* that disturbs monotonic nature of entropy of a constraint. We go beyond the identification of nasty constraints to pin-point the singleton to restore normal behaviour of entropy. On termination, the algorithm guarantees either an exact or an approximate solution depending upon the number of constraints the solution violates. We demonstrate experimentally that solution to general IA networks can be efficiently obtained in time polynomial in the size of the network with the success rate of 95% contrary to exponential exact algorithm.

Keywords: Constraint satisfaction problem, Approximation algorithm, Interval algebra

1 Introduction

Constraint Satisfaction Problems (CSP) are in general NP-hard class [6]. On the other hand, CSPs have numerous applications in almost all branches of engineering. There have been several attempts to devise solution techniques for CSPs. One approach is to characterize tractable subclasses and to provide polynomial-time algorithm for solving such instances. Another approach is to devise good heuristics and search strategies. In order to understand the distribution of hard instances, there have also been studies on identifying values of critical parameters which lie between the easy instances of under-constrained and of over-constrained instances. In this paper, we attempt to characterize the hardness of problem instances in a different manner. One wonders whether there are certain nasty constraints in an instance of CSP that is possibly the reason for hardness. And if so is the case, this paves the way to devise approximation scheme to solve hard instances by settling these nasty constraints. The efficiency of such an approximation method lies in settling very small number of nasty constraints to obtain a solution in polynomial time.

In this paper we study this aspect in the context of qualitative temporal CSP, namely Allen's framework [1] IA. We characterize a reason for late solution for prob-

© A. Gelbukh, R. Monroy. (Eds.)
Advances in Artificial Intelligence Theory
Research on Computing Science 16, 2005, pp. 55-64

lems due to presence of nasty constraints. The major contribution of this paper can be highlighted as follows.

In this paper, we introduce a new polynomial time complete method for determining an approximate solution of IA network. We start with a path consistent IA network. An iterative transitive closure algorithm such as weighted path consistency assigns highest weight to an atomic relation in a label that has maximum likelihood to be a feasible relation. Intuitively, when the highest weight relation on the edge agrees with the relation with highest weight in the constraint computed by averaging along all paths, this relation is the best possible candidate feasible relation in the constraint. A conflict occurs when the highest weight relation is not the same as that from the paths. This includes following two possibilities: (a) Highest weight relation on the edge is present in the averaged constraint, but with a lower weight, (b) Highest weight relation on the edge is not present at all in the averaged constraint. Following our intuition, in order to forcefully make the two agree, either we raise the lower weight of an existing relation to become the highest weight or we introduce a new atomic relation with highest weight. We term the constraint that exhibit the property of introducing new singletons as highest weight relation to resolve the conflict as *nasty constraint.* This adjustment of weights helps us to reduce the conflicts as and when they appear in an iteration. In case of conflicts, the solution may or may not violate any constraint. This helps in computing an approximate and early solution for hard instances. We prove that presence of nasty constraints in hard instances is responsible for preventing entropy of constraint from decreasing monotonically. Introducing the required singletons, restores the monotonic decrease in the subsequent iterations. Experiments reveal this method solves general IA networks by violation of small fraction of constraints.

In Section 2, we present IA framework and related work. In Section 3, we summarize weighted path consistency. In Section 4, we introduce the concept of entropy for weighted IA network with preliminary experiment. In Section 5, we propose that nasty constraints reflect the hardness of any given problem instance in IA with theoretical justification in Section 6, which contains the main result of this paper, a polynomial time complete algorithm for approximate solution of hard TCSP. We report experimental analysis in Section 7. Section 8 contains conclusions.

2 Interval Algebra and Related work

IA defines thirteen atomic relations that can hold between any two time intervals, namely before(b), meet(m), ovelap(o), start(s), during(d), finished-by(fi), equal(eq), finish(f), contain(di), started-by(si), overlapped-by(oi), meet-by(mi) and after(bi) [1]. In order to represent indefinite information, the relation between two intervals is a disjunction of the atomic relations. Reasoning for the complete interval algebra is known to be NP-hard [22]. Traditional solution techniques for temporal and spatial domains are either based on complete[8, 11, 18, 20] or partial instantiation strategies[21]. So far there is no complete method based on complete instantiation strategy for approximate solutions for qualitative TCSPs.

Research in phase transition is investigated [9, 7, 3, 5, 4] to study instance hardness. In the context of IA network, it is not possible to have any estimate of number of solutions. So far we roughly know the hard instances exist for a combination of parameters. The entropy based analysis of nasty constraints looks to be promising enough to open up a new study in this direction for qualitative CSPs. Basically to study a discrete problem consisting of only disjunctions, we are translating it to a continuous domain by adapting a weighted formalism.

3 Weighted path consistency

In this paper, we use weighted path consistency algorithm as proposed in [13], [2]. In a weighted IA network *W(N)* each constraint is represented as a 13-dimensional weight vector $W_{ij} \in R^{13}$ such that $0 \le W_{ij}^{m} \le 1$, $1 \le m \le 13$, $\Sigma W_{ij}^{m} = 1$. W_{ij}^{m} denotes the weight of the atomic relation IA_m in the constraint between variables *i* and *j*. The value 0 for W_{ij}^{m} implies IA_m is absent in the disjunction. We call each W_{ij} as *weighted constraint*. Given an IA network *N*, we obtain the corresponding weighted network by assigning equal weights to all the atomic relations present in a constraint. We represent the IA-composition table [6] as a 3-dimensional binary matrix **M**, such that $\mathbf{M}_{ijm} = 1$ if and only if the atomic relation IA_m belongs to the composition of the atomic relations IA_i and IA_j. The composition of two weighted relations W_{ik} and W_{kj} resulting in a relation $W_{ij}(k)$ is denoted as $W_{ik} \otimes W_{kj}$. The intersection of two weighted relations W_{ij} and V_{ij} is denoted as $U_{ij} = W_{ij} \cap V_{ij}$, defined as follows [13]:

$$W_m^{ij}(k) = \frac{\sum_u \sum_v M_{uvm} W_u^{ik} W_v^{kj}}{\sum_m \sum_u \sum_v M_{uvm} W_u^{ik} W_v^{kj}}, \quad 1 \le m \le 13. \qquad U_m^{ij} = \frac{W_m^{ij} V_m^{ij}}{\sum_m W_m^{ij} V_m^{ij}}, \ 1 \le m \le 13$$

We follow a slightly different approach for computing the averaged constraint along the paths. For each edge, first we compute the non-zero normalized average of all vectors that are computed by path-wise composition. This averaged vector is intersected with the edge vector followed by normalization. We use intersection operator only once which reduces numerical computations. The weighted path consistency algorithm, unlike the conventional path consistency, modifies only the weights of constraints. The atomic relations with higher weights are more favorable to be the feasible ones, whereas those with smaller weights are less likely to participate in a solution. There will be no occasion when the weight values in the vectors will stop changing unless it is a network only with singleton labels(trivial case).

4 Entropy of IA network

In this section, we introduce the concept of entropy for IA network in the context of weighted formalism. In [14, 15, 16], the three properties of measures on entropy given in [19] are generalized. The Renyi's quadratic entropy (RQE) is given as $-\log \sum_m (W_m^{ij})^2$. In the context of minimizing entropy, for the sake of convenience, *log* in the above expression is normally dropped. For the present study, we loosely define entropy to be without *log* [23].

Definition 1: Entropy of a weighted constraint W^{ij} is defined as follows

$$E_w^{ij} = -\sum_m (W_m^{ij})^2 \quad where \quad 0 \le W_m^{ij} \le 1 \; and \; \sum W_m^{ij} = 1$$

Definition 2: Entropy of a weighted network is defined as follows

$$E_N = \sum_{ij} E_w^{ij}$$

The least entropy of a constraint corresponds to a singleton relation and the highest entropy is when it has non-atomic relations with equal weights. Path consistency algorithm indeed prevents the entropy of the constraint network from increasing.

Theorem 1: For a given network *N*, enforcing path consistency does not increase E_N .

We illustrate this with the help of a simplex triangle (Figure 1) for a constraint with a maximum of three atomic relations. The three vertices **C**, **D** and **E** are the lowest entropy points that correspond to the three possible singleton labels for the constraint. The centre **A** corresponds to the highest entropy corresponding to equal weights for the three relations. The contours represent states with equal entropy. The entropy state at an edge indicates a conflict between the two singleton labels.

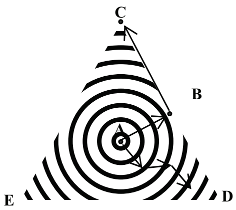


Figure 1. Simplex Triangle

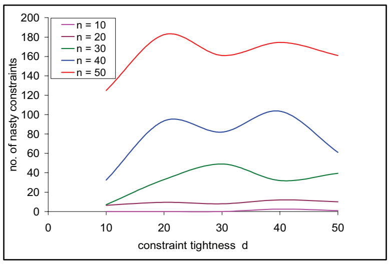


Figure 2. No. of constraints with fluctuating entropy for M(n,d,7) for known consistent problems.

In the conventional path consistency, we move from A to C in one step or from A to B and then possibly to C. In the event of a solution, the search ends at a vertex else it stops at either A or at B. Ideally, any search technique should choose a descending path from A to one of the vertices, say D (Figure 1). We have experimented initially with convex IA networks. For our experimental study, we generate random instances based on three parameters, namely network size(n), constraint tightness(d) and label size(t) [12], [13]. We experimented with 200 instances for each value of n in the range [10,100]. There was not a single instance out of randomly generated 3800 convex problem instances with any fluctuation in the entropy. We repeat the same experiment for general IA networks that are known to be consistent. We find that unlike the convex case, the entropy values of some constraints increase after an initial decrease, but again continue to decrease until stabilization i.e non-monotonic behaviour. Thus for convex network, the entropy value for every edge takes a descending path from the centre of the simplex to a boundary (Figure 1), a monotonic behaviour. On the other hand, the trajectory of entropy value of any edge in a non-convex network need not be a descending path but a longer trajectory path. The search begins at the centre towards a vertex, but moves along the periphery of a contour with smaller entropy.

The number of constraints with fluctuations in entropy is very high for the problem instances in the hard region [12]. Figure 2 depicts the number of such constraints in instances with $t = 7$, n in the range [10,50] and d in [10, 50]. The peaks in the graphs (Figure 2) are the regions of hard instances (one to one correspondence of peaks is not possible due to the different model for generating problem instances used in this study). This has motivated us to study the behaviour of entropy in order to identify the cases when entropy increases. This study narrows down to basic fundamentals of multiplication of two vectors. The difference between entropy of a pair of vectors consisting of same number of non-zero entries, depends on the relative order of their highest value. When the number of non-zero entries in the two vectors is not same, then one cannot conclude clearly as which of the two will have a higher value of entropy. It depends on the relative distribution of the values within the vector. We formalize these observations as following results:

Theorem 2: Given two normalized vectors U and V with same no. of non-zero components, i.e. $\mathrm{nz}(U) = \mathrm{nz}(V)$, $|\mathrm{E}(U)| \geq |\mathrm{E}(V)|$ iff $\max(U) \geq \max(V)$, where $\max(U)$ is the component with highest value.

Theorem 3: Given two normalized vectors U and V such that $\mathrm{nz}(V)>\mathrm{nz}(U)$, then $\max(V)>\max(U)$ is not a sufficient condition for $|\mathrm{E}(V)| < |\mathrm{E}(U)|$.

5 Nasty Constraints

In this section, we identify a new property of weighted IA constraints called *nastiness* that is responsible for a difficulty in computing solution of a problem instance. Suppose W_{ij} is the weighted constraint on the edge *(i,j)* and W is the averaged constraint obtained from all possible paths using the weighted composition operator as explained in section 2. W_{avg} is the constraint obtained by weighted intersection of W_{ij} and W at the end of the current iteration of weighted path consistency. We study the impact of replacing W_{ij} by W_{avg} in terms of weights of atomic relations that will increase or decrease with the help of inner product of vectors.

Lemma 1: Given two weighted IA constraints $U = [u_i]$ and $V = [v_i]$, the normalization factor $\lambda = \sum u_i v_i$ will satisfy the conditions, $u_{min} \leq \lambda \leq u_{max}$ and $v_{min} \leq \lambda \leq v_{max}$ where $u_i \in [u_{min}, u_{max}]$, $v_i \in [v_{min}, v_{max}]$.

Theorem 4: $W_{avg}[\mathrm{argmax}(W_{ij})] > W_{ij}[\mathrm{argmax}(W_{ij})]$
iff $\mathrm{argmax}(W_{ij}) \in W_{high}$ where $\forall p \in W_{high}$, $W[p] \geq \lambda$, $\lambda = \sum u_i v_i$.

The normalization factor divides the vector W into two halves, the relations with weights greater than normalization factor will increase if W_{avg} replaces W_{ij}. In other words, if the highest weight relation on the edge is among the higher weight relations in the averaged constraint along the paths, then its weight is guaranteed to increase further. Contrary to this, when the highest weight relation on the edge is not the highest weight relation in the averaged constraint, a conflict takes place. Whether this conflict will lead to a decrease in the weight of the highest weight relation on the edge by replacing W_{ij} by W_{avg}, is an obvious consequence of the above theorem. We formalize this condition as the following lemma.

Lemma 2: If $\text{argmax}(W_{ij}) \notin W_{high}$, then $W_{avg}[\text{argmax}(W_{ij})] < W_{ij}[\text{argmax}(W_{ij})]$.

Our premise is that as the weighted path consistency algorithm iterates, the weights in the network are adjusted based on the influences of the weights of the edges along the paths. Thus in an ideal situation (for eg a convex network), above lemma should not be satisfied at all. There are two possibilities here, the highest weight relation may exist in the averaged constraint with a lower weight or may be absent i.e. a weight of zero. In the first case, we solve the conflict by forcing the weight in the constraint resulting after intersection to be the highest value such that it becomes the highest weight relation for the next iteration. In the second case, as mentioned in the earlier section, the entropy of the constraint may or may not increase. In the latter case, a new atomic relation is forced to dominate other weights in the next iteration. For the cases when the next iteration highest weight value is less than the highest weight value in the current iteration, entropy will increase otherwise it will continue to decrease. We formalize these observations to introduce a concept of nasty constraints for IA networks.

Definition 3: A weighted IA constraint is said to be a *nasty constraint* if it satisfies either of the following conditions:

(a) If $\text{argmax}(W_{ij}) \notin W_{high}$ and atomic relation at $(\text{argmax}(W)) \notin R_{ij}$.
(b) If $\text{argmax}(W_{ij}) \notin W_{high}$ and atomic relation at $(\text{argmax}(W)) \in R_{ij}$.
and $W_{avg}[\text{argmax}(W)] < W_{ij}[\text{argmax}(W_{ij})]$

where R_{ij} : constraint on edge *(i,j)* in the current iteration of weighted path consistency.

By the study of entropy of weighted IA constraints in the previous section, it is obvious that by the very definition of nasty constraint, entropy of a nasty constraint will increase when either of the above two conditions are satisfied. We formalize this consequence as following result.

Theorem 5: Entropy of a nasty constraint does not decrease monotonically over iterations of weighted path consistency.

A vector that is initially generated with all the components with equal values, this will correspond to maximum entropy of the vector. If the same vector is subjected to some operations in an iterative manner such that the value of one of the components goes on dominating all others, the entropy of this vector will go on decreasing assuming number of non-zero components do not change. A stage will come, beyond which entropy cannot decrease further and hence stabilizes. We exploit this observation in the next section as the termination condition of the algorithm proposed in this paper.

6 Approximate solution for IA networks

In this section, we propose a method to determine an approximate solution for IA networks based on our foregoing analyses. We propose an algorithm to identify nasty constraints in weighted IA networks and settle these to compute an early solution as shown in the pseudocode in Table 3.

```
compute_approx_solution(W(N))
        Output: A singleton network τ that is a solution
        while no solution   weighted_path_consistency_iteration(W(N)) enddo
weighted_path_consistency_iteration(W(N))
        ∀W_ij     ∀k = 1 to n, such that k≠i and k≠j
                          W(k) = W_ik ⊗ W_kj
                  W ← normalized non-zero average over W(k)
                  λ = Σ w[p]w_ij[p], p = 1 to 13
                  W_avg[i,j] ← W ∩ W_ij
                  {where ⊗ and ∩ are weighted composition and intersection operators}
                  partition W such that W=W_high∪W_low, W_high∩W_low=∅,
                                        ∀p∈W_high, W[p]≥λ, ∀q∈W_low, W[q]<λ
                  if (argmax(W_ij) ∉ W_high)
                      if (IA(argmax(W_ij))∉R_ij) mark (i,j): nasty constraint endif
                      if (IA(argmax(W_ij))∈R_ij) and (W_avg[argmax(W_ij)]<W_ij[argmax(W_ij)])
                          mark (i,j) as a nasty constraint
                      endif
                      W_avg[argmax(W)] = 1.0
                      renormalize W_avg
                  endif
                  Replace ∀(i,j)   W_ij ← W_avg(ij)
                  ∀(i,j)  τ_ij ← atomic relation at argmax(W)
                  if τ is path consistent then solution found
                          ∀(i,j)   if τ_ij ∉ C_ij then constraint is violated endif
                          where C_ij is the disjunctive constraint in the IA network N
                  endif
```

Table 3. approximate solution algorithm.

Clearly *compute_approx_solution* is of $O(n^3T)$ complexity, if we assume that T number of iterations of weighted path consistency are executed to compute a solution. As per our foregoing analyses in the previous section, this algorithm captures those constraints as nasty constraints for which entropy fluctuates. It is observed that there are some more constraints that are not the nasty constraints, but still the highest weight relation along the paths is forced to become highest on the edge. These are those constraints for which a conflict takes place and the highest on the edge is not absent along the path, but has smaller weight. We term all the constraints (including nasty constraints) where any time this type of adjustment of weights takes place as *approximated constraints*(AC). The algorithm starts with the state of highest entropy for all the constraints, that corresponds to the starting point when all the atomic relations in a constraint are assigned equal weights. As the weighted path consistency algorithm iterates, *compute_approx_solution* ensures that entropy of every constraint to decrease monotonically. In the later iterations, weight of an atomic relation dominates others, leading to the state of least entropy beyond which a bounded variable like entropy (with a minimum value of -1) cannot decrease. Over iterations of weighted path consistency, our algorithm reduces the number of inconsistent triplets in the singleton network by forcing the highest weight relation on the edge to agree with the one with maximum support along the paths. We claim that on termination, it will compute a solution. The solution may be an exact one for easy instances and an approximate one for hard instances. Thus our method is a complete method for determining approximate solution for IA networks.

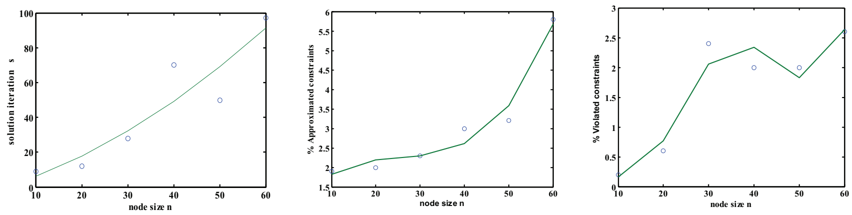
Theorem 6: *compute_approx_solution* is a complete algorithm.

7 Experimental analysis

The objective of the experimental analysis is essentially to confirm our theoretical analyses as discussed in the previous sections. Realizing the algorithm is complete, we attempt to determine the instances that are known to be consistent and completely cover both easy as well as hard problem regions of IA networks. The experiments are conducted on Windows based PC with 2GHz clock speed, 512 RAM and Visual C++ environment. We have experimented with 480 instances of known consistent general IA networks with n in the range [10,60]. The graphs shown in Figure 4 indicate the performance of our method.

The model for instance generation is same as that proposed in [13]. The iteration at which a consistent scenario is obtained, (s) is noted and average of these is taken for each combination of n and d. This approach gives an empirical estimate of average number of iterations required to get a solution for general IA networks. We make use of statistical regression models to analyze empirical results at arrive at the best-fitting curve. Figure 4(a) shows that the solution iteration depends on the constraint tightness. Figures 4(b) explains that higher the number of approximated constraints higher is the number of violated constraints. Our method is able to solve 100% of the problems for n in the range [10,40]. Two instances for 40 nodes and five problems in 60 nodes set of problems are left without a solution, i.e. 95% of success rate. With experience, we say that this 5% of failure is due to numerical errors.

Any comparison of this method with backtrack algorithm will not be in place. We feel that comparison of two methods that give different types of solutions does not help us in this context. However, an outright advantage of our method can be simply seen by the fact that for 50 and 60 node problems, backtrack is known to take exponentially high computation time, where as our method gives the solution in a maximum computation time of 80 minutes, which is equivalent to a maximum of 1000 iterations of weighted path consistency algorithm. This method is able to solve even hard problems in reasonable time despite a large number of nasty constraints.



(a) **(b)** **(c)**

Figures 4. Linear cubic regression models.
(a) $s = 0.0136 + 0.7529n - 2.6n^2$
(b) $AC = 3.83 + 0.7475n + 0.0480n^2 - 0.0011n^3$
(c) $VC = 0.73 + 0.16n - 0.00594n^2 + 0.0001n^3$

8 Conclusions

The present work introduces a new paradigm for TCSP using entropy-based interpretation of IA as against the known method backtrack. We provide an insight into the well-known fact that convex networks are easy to solve. General IA problems with relations not belonging to any of the tractable classes are solved with help of a complete method. We provide here a linear time algorithm that captures the hardness of the problem in terms of nasty constraints, exploiting structure of individual problems. Our algorithm computes approximate solution for hard problems in polynomial time with exact solution a special case. In the process of handling the conflicts, the link with the original problem is not lost. It is possible for an interactive choice of nasty constraints to be settled, that may be crucial to the problem. It is possible to keep track of iteration-wise resolved atomic relations. User can analyze the impact of avoiding or choosing a new atomic relation. We propose to extend this study to propose PTAS with approximation bounds for general IA networks and overconstrained problems.

References

1. Allen: Maintaining knowledge about temporal intervals. *Communication of the ACM*, 26(11): 832-843, 1983.
2. Bhavani, S. D. and Pujari, A. K.: EVIA-Evidential Interval Algebra and Heuristic backtrack-free algorithm, *Constraints*, 9 (3), 2004, 193.
3. Boukeas, G., Halasis, G. Zissimopoulos, V. and Stamatopoulos, P.: Measures of Intrinsic Hardness for Constraint Satisfaction Problem Instances. LNCS 2932, 184-195, 2004.
4. Clark, D.A., Frank, J. Gent, I.P., Macintyre, E., Tomov, N. and Walsh, T.: Local Search and the Number of Solutions. *Proc. of CP-96*, Springer(1996), 119-133.
5. Crutchfield, J. Feldman, D.: Regularities Unseen, Randomness Observed: Levels of Entropy Convergence. Chaos 13(2003) 25-54.
6. Dechter: *Constraint Processing*. Morgan Kaufmann Publishers, San Francisco, USA, 2003.

7. Gent, I.P., MacIntyre, E., Prosser, P., Walsh, T.: The Constrainedness of Search. In: AAAI/IAAI. Volume 1. (1996) 246-252.
8. Gerevini, A. and Renz, J. Combining topological and qualitative size constraints for spatial reasoning. *Proc. of the 4th International Conference on Principles and Practice of Constraint Programming* 1998, Pisa, Italy.
9. Hogg, T., Huberman, B. Williams, C.: Phase Transitions and the Search Problem. Artificial Intelligence 81 (1996) 1-15.
10. Larrosa, J., and Schiex, T.: In the quest of the best form of local consistency for weighted CSP. In *Proc. of the 18th IJCAI* (Acapulco, Mexico, 2003), 239–244.
11. Ligozat, G.: A new proof of tractability for ORD-Horn relations. *Proceedings of AAAI-96*, 395-401.
12. Nebel: Solving hard qualitative temporal reasoning problems: Evaluating the efficiency of using the ORD-Horn class. *Constraints*, 1(3):175-190, 1997.
13. Pujari, A.K., and Adilakshmi, T.: A Novel Heuristic to Solve IA Network by Convex Approximation and Weights. In *Proc. of the 8th PRICAI* (Auckland, New Zealand, August 2004), 154-163.
14. R´enyi, A.: On measures of entropy and information. Selected Papers of Alfred R´enyi, 2(180):565–580, 1976a.
15. 18. R´enyi, A.: On the foundations of information theory. *ibid*, 3(242):304–318, 1976b.
16. 19. R´enyi, A.: Some fundamental questions of information theory. *ibid*, 2(174):526–552, 1976c.
17. Rossi, F., Venable K. B., Khatib, L., Morris, P. and Morris, R.: Two solvers for tractable temporal constraints with preferences. *Proc. AAAI 2002 Workshop on preferences in AI and CP Edmonton,* Canada.
18. Selman, B., Levesque, H. and Mitchell, D.: A new method for solving hard satisfiability problems. In Proceedings of the *Tenth National Conference on Artificial Intelligence*, 440-446, 1992.
19. Shannon, C. E. and Weaver, W.: The Mathematical Theory of Communication. Univ of Illinois Pr., 1963. ISBN: 0–252–72548–4.
20. Thornton, J., Beaumont, M., Sattar, A. and Maher, M. J.: Applying Local Search to Temporal Reasoning. *TIME,* 2002: 94-99.
21. van Beek and Manchak 1996: The design and experimental analysis of algorithms for temporal reasoning. *JAIR 4*:1-18.
22. Vilain, M. and Kautz, H.: Constraint propagation algorithms for temporal reasoning. In *Proc. Fifth National conference on Artificial Intelligence.* 377-382, Philadelphia, 1986.
23. Zitnick III, C. L.: Computing Conditional Probabilities in Large Domains by Maximizing R´enyi's Quadratic Entropy. PhD Thesis: CMU-RI-TR-03-20, 2003.