# TagSNP selection using weighted CSP and russian doll search with tree decomposition

David Allouche, Thomas Schiex, Simon de Givry, Martin Sanchez

# TagSNP selection using Weighted CSP and Russian Doll Search with Tree Decomposition

D. Allouche, S. de Givry, M. Sanchez, T. Schiex

UBIA, UR 875, INRA, F-31320 Castanet Tolosan, France.
{allouche,degivry,msanchez,tschiex}@toulouse.inra.fr

**Abstract.** The TagSNP problem is a specific form of compression problem arising in genetics. Given a very large set of SNP (genomic positions where polymorphism is observed in a given population), the aim is to select a smallest subset of SNPs which represents the complete set of tagSNP reliably. This is possible because strong correlations existing between neighboring SNPs. Typically, besides minimizing the tagSNP set size (mostly for economical reasons), one also seek a maximally informative subset for the given size, generating different secondary criteria.
This problem, which is also closely related to a set covering problem, can be simply described as a weighted CSP. We report here our experiments with human tag SNP data using a recently designed WCSP algorithm combining the "Russian Doll Search" algorithm with local consistency for cost functions and an active exploitation of the problem structure, through a tree decomposition of the problem.

## Introduction

In bioinformatics, uncertainty and variability are usually ubiquitous and combinatorial problems modelling biological objects or phenomenon usually involve a combination of a large number of local uncertainties or correlations which must be incorporated in a global criteria. It is therefore not surprising to see that graphical models (HMM, Bayesian nets, (conditional) Markov random fields...) are massively used in bioinformatics.

The constraint satisfaction problem and its implementation as constraint programming languages represent a deterministic variant of graphical models which allows to represent combinatorial problems on discrete variables. Optimization (if any) is often considered as a second order target that can be reached by explicitly modeling a cost variable which is then iteratively bounded until an optimum is reached. This however requires the use of global constraints that cover all variables, thereby hindering all the problem structure.

In this paper, we consider the TagSNP problem, a specific form of lossy compression problem arising in genetics. Given a very large set of SNP (genomic positions where polymorphism is observed in a given population), the aim is to select a smallest subset of SNPs which represents the complete set of tagSNP reliably. This problem can be naturally defined as a variant of the set covering problem. By modeling the problem as a weighted CSP (or cost function network),

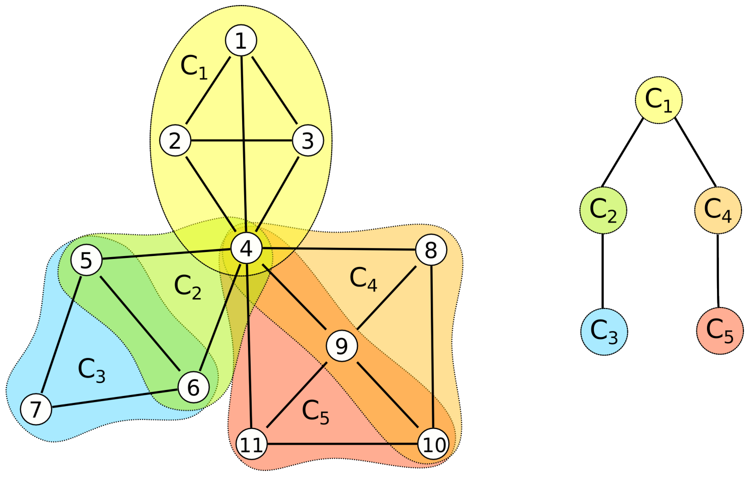

**Fig. 1.** The graph of a WCSP and its tree decomposition.

we get a model which still offers opportunities to exploit the problem structure described through a tree-decomposition of its graph.

Specifically, this enables the fruitful application of a recent algorithm [11] called "Russian Doll Search with Tree Decomposition" that exploits problem structure for solving cost function networks. As the BTD algorithm [12], this algorithm is based on the identification of conditionally independent subproblems, which are solved independently, using a lower bound based on local consistency, and whose optimum is cached during Branch and Bound search. The time complexity of the algorithm obtained is only exponential in the largest subproblem size instead of the global problem size. These conditionally independent subproblems are called clusters ad form a tree. The intersection between two clusters, if not empty, is called the separator between these two clusters (see Fig. 1 for an illustration). Such a decomposition can be obtained by computing a so-called tree-decomposition of the problem. The main novelty of our algorithm lies in the incorporation of a "Russian Doll Search" like approach enabling the computation of stronger initial local bounds by inductively solving a relaxation of each subproblem identified in the tree-decomposition.

The algorithm obtained, BTD-RDS, generalizes both RDS and other tree-decomposition based algorithms such as BTD or AND-OR Branch and Bound. As BTD, it uses a restricted dynamic variable ordering which must be compatible with the tree decomposition exploited : a variable from a cluster cannot be instantiated before variables from its parent cluster.

This algorithm has been applied to radio link frequency assignment problem instances defined in the CELAR benchmark [2], closing a very hard frequency assignment instance which has been open for more than 10 years. It has also been used to tackle a problem from bioinformatics, defining a new benchmark for constraint-based approaches : the tagSNP selection problem.

In this paper, we will present this problem with associated instances defining benchmarks and the main results obtained in our experiments.

## 1 Modeling the TagSNP problem

TagSNP selection occurs in genetics and polymorphism analysis. Single nucleotide polymorphisms, or SNPs, are DNA sequence variations that occur when a single nucleotide (`A`,`T`,`C`,or `G`) in the genome sequence of an individual is altered. For example a SNP might change the DNA sequence `AAGGCTAA` to `ATGGCTAA`. For a variation to be considered as a SNP, it must occur in at least 1% of the population. There are several millions SNPs in the 3 billions nucleotides long human genome, explaining up to 90% of all human genetic variation. SNPs may explain a portion of the heritable risk of common diseases and can affect response to pathogens, chemicals, drugs, vaccines, and other agents. The TagSNP problem is a sort of lossy compression problem which consists in selecting a small subset of SNPs such that the selected SNPs, called tag SNPs, will capture most of the genetic information. The goal is to capture a maximally informative subset of SNPs to make screening of large populations feasible [6]. From the combinatorial point of view the TagSNP problem is equivalent to a set covering problem (NP-hard) with additional quadratic criteria.

By sampling a first relatively small population, it is possible to compute a link (correlation) measure $r^2$ between each pair of SNPs. A tag SNP is considered as representative of another SNP if the two SNPs are sufficiently linked. The simplest TagSNP problem is to select a minimum number of SNPs (primary criteria) such that all SNPs are represented. This is captured by the fact that the $r^2$ measure between the two SNPs is larger than a threshold $\theta$ (often set to $\theta = 0.8$ [3]). We therefore consider a graph where each vertex is a SNP and where edges are labelled by the $r^2$ measure between pairs of nodes. Edges are filtered if their label is lower than the threshold $\theta$. The graph obtained may have different connected components. The TagSNP problem then reduces to a set covering problem on these components. This simplest variant has been studied in [1] where it is solved using the d-DNNF compiler `c2d` with good results.

In practice the number of optimal solutions may still be extremely large and secondary criteria are considered by state-of-the-art tools such as FESTA [9]. Between tag SNPs, a low $r^2$ is preferred, to maximize tag SNP dispersion. Between a non tag SNP and its representative tag SNP, a high $r^2$ is preferred to maximize the representativity. To optimize these criteria, FESTA uses two incomplete algorithms, the simplest is called FESTA-greedy, and it uses a simple reedy approach. The second, called FESTA-hybrid combined the greedy approach with a limited exhaustive approach.

For a given connected graph $G = (V, E)$, we build a binary weighted CSP with integer costs capturing the TagSNP problem with the above secondary criteria. For each SNP $i$, two variables $i_s$ and $i_r$ are used. $i_s$ is a boolean variable that indicates if the SNP is selected as a tag SNP or not. The domain of $i_r$ is the set of neighbors of $i$ together with $i$ itself. It indicates the representative tag SNP which covers $i$. For a SNP $i$, hard binary cost functions (with 0 or infinite costs) enforces the fact that $i_s \Rightarrow (i_r = i)$. Similar hard cost functions enforce $(i_r = j) \Rightarrow j_s$ with neighbor SNPs $j$ in $G$. A unary cost function on every variable $i_s$ generates an elementary cost $U$ if the variable is *true*. The resulting weighted CSP captures the set covering problem defined by TagSNP.

To account for the representativity, a unary cost function is associated with every variable $i_r$ that generates cost when $i_r \neq i$. In this case, the cost generated is $\lfloor 100.\frac{1-r^2_{i,i_r}}{1-\theta} \rfloor$. For dispersion between SNPs $i$ and $j$, a binary cost function between the boolean $i_s$ and $j_s$ is created which generates a cost of $\lfloor 100.\frac{r^2_{ij}-\theta}{1-\theta} \rfloor$ when $i_s = j_s = \mathit{true}$ The resulting WCSP captures both dispersion and representativity. In order to keep these criteria as secondary, we just use a large enough value for $U$ (the elementary cost used for tag selection).

This problem is similar to a set covering problem with additional binary costs. Such secondary criteria are ignored by [1]. Here, `c2d` yields a compact compiled representation of the set of solutions of the pure set covering problem, but the number of solutions is so huge (typically more than billions) that applying the second criteria on solutions generated by `c2d` would be too expensive. A direct compilation of the criteria in the d-DNNF does not seem straightforward and would probably necessitate a Max-SAT formulation as the authors acknowledge in their conclusion.

## 2 Experiments

The algorithms tested (BTD and RDS-BTD) have been implemented in `toulbar2` C++ solver[1]. Note than when the tree decomposition used reduced to a single cluster, BTD is equivalent to a Depth First Branch and Bound algorithm.

The *min domain / max degree* dynamic variable ordering, breaking ties with maximum unary cost, is used inside clusters (BTD and RDS-BTD) and by DFBB. The dynamic variable ordering heuristic is modified by a conflict back-jumping heuristic as suggested in [8]. EDAC local consistency is enforced [5] during search. Tree decompositions are built using the Maximum Cardinality Search (MCS [10]) heuristic, with the largest cluster used as root. A variable ordering compatible with the rooted tree decomposition used is used for DAC enforcing [4].

All the solving methods exploit a binary branching scheme. If $d > 10$, the *ordered* domain is split in two parts (around the middle value), else the variable is assigned to its EDAC fully supported value or this value is removed from the domain. In both cases, it selects the branch which contains the fully supported value first, except if a previous solution is available (the corresponding value is used in this case). Reported CPU times correspond to finding the optimum and proving its optimality.

The instances we considered have been derived from human chromosome 1 data provided by courtesy of Steve Qin [9]. Two values, $\theta = 0.8$ and 0.5 have

[1] http://mulcyber.toulouse.inra.fr/projects/toulbar2, version 0.8.

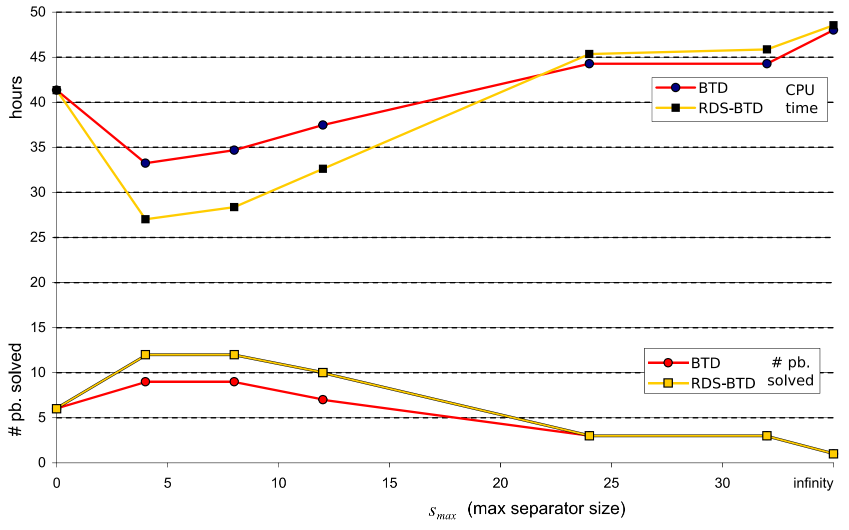

**Fig. 2.** CPU-time and number of solved instances for different values of the maximum separator threshold ($s_{max}$).

been tried. For $\theta = 0.8$, a usual value in tag SNP selection, $43,251$ connected components are identified among which we selected the 82 largest ones. These problems, with 33 to 464 SNPs, define WCSP with domain sizes ranging from 15 to 224 and are relatively easy. Solving them to optimality selects 359 tag SNPs in 2h37' instead of 487 in 3' for FESTA-greedy (21% improvement) or 370 in 39h17' for FESTA-hybrid (3% improvement, 15-fold speedup).

To get more challenging problems, we lowered $\theta$ to 0.5. This defined $19,750$ connected components, among which 516 are not solved to optimality by FESTA. We selected the 25 largest one. These problems, with 171 to 777 SNPs have graph densities between 6% and 37%. They define WCSP with max domain size ranging from 30 to 266 and include between 8000 to $250,000$ cost functions. The decomposability of these problems, estimated by the ratio between the treewidth of the original MCS tree-decomposition (without any cluster merging, i.e. with $s_{max} = +\infty$) and the number of variables varies from 14% to 23%.

In theory, algorithms exploiting tree-decompositions are only exponential in the maximum cluster size and the ideal tree-decomposition that should be used is therefore a tree decomposition minimizing the size of the largest cluster (also called the treewidth or induced width of the problem). In practice however, when a Branch and Bound like approach such as BTD is used, small separators are also attractive because the worst-case space complexity of the algorithm is exponential in the separator size. Even if this space complexity is usually far from being reached because of the pruning induced by lower bounds, small separators are also attractive because the removal of large separators creates larger clusters that gives more freedom to dynamic variable ordering. There is

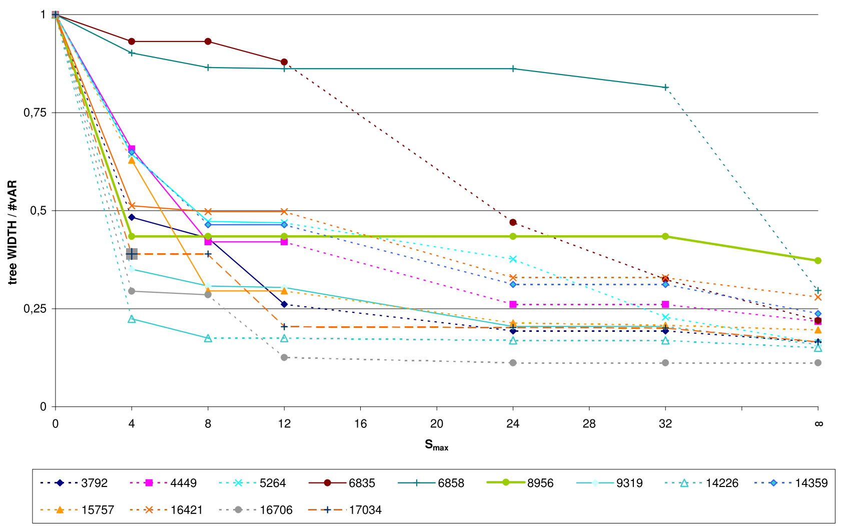

**Fig. 3.** Evolution of treewidth normalized by variable number for different maximum separator size threshold $s_{max}$.

therefore a compromise to reach : small clusters are desirable but large separators should be avoided. Starting from an original tree-decomposition, it is always possible to reach a decomposition with a maximum separator size below a given threshold $s_{\max}$ by just merging any pair of clusters which has a separator of size above $s_{\max}$ [7].

All the problems have been tackled with an initial upper bound found by FESTA-greedy on 2.8 Hz CPU with 32 GB RAM. To better show the importance of bounded separator size (using $s_{max}$), we considered values ranging from 0 (DFBB), 4, 8, 12, 24, 32 to $+\infty$ for both BTD and RDS-BTD. We report both the number of problems solved within a 2-hour limit per instance and the total amount of CPU time used (an unsolved instance contributes for 2 hours).

Using $s_{max} = 4$, our implementation improves the compression ratio of FESTA-greedy by 15% (selecting 2952 tag SNPs instead of 3477 for the $516 - 13$ solved instances). Note that the differences in CPU time between BTD and RDS-BTD would increase if a larger time limit had been used. From a practical viewpoint, the criterion of the TagSNP problem could be further refined to include : sequence annotation information (*e.g.* preferring tag SNPs occurring in genes), and measures between triplets of markers as proposed in [1] (SNPs covered by a pair of tag SNPs). The good performances of RDS-BTD may allow to tackle this more complex problem with realistic $\theta = 0.8$.

Figure 3 presents the evolution of the ratio between the tree width and the problem size for different values of $s_{max}$. All instances ($\theta = 0.5$) solved at least once during the experimentation are shown in this graph.

Problems which are successfully solved (with an optimality proof) are materialized as continuous line,while dashed lines represent failures. The instance 17034, which could be solved only for $s_{\max} = 4$ is represented by a gray square.

The Figure 3 allows to materialize the 12 instances which could be solved for $s_{max} = \{4, 8\}$. This underlines the fact that the nature of each solved instance may be different depending of the $s_{max}$ value. For example, instance 17034 is only solved for $s_{max} = 4$, which is balanced by the resolution of instance 14359, solved only on the interval 8 - 12 of $s_{\max}$.. Overall, on the whole range of variation of $s_{\max}$, 13 instances are solved but only 12 can be solved for an optimal choice of $s_{max} = 4, 8$. This may plead for an adaptive choice of the parameter $s_{max} = 4, 8$. Moreover, a significant decrease of the normalized treewidth is often correlated with the occurrence of a successful resolution.

Indeed, for most of the tested instances, reflecting the structure of the problems, the curves show an important decrease for $s_{max} \in \{0, 8\}$, which is very likely one of the significant events explaining the high number of successful resolution in this range. The decrease of the treewidth is then lower. It gradually reaches ($s_{max} > 12$), a plateau with a gentle slope, asymptotically reaching the ratio corresponding to a full decomposition of the problem (without cluster merging). In the experimentation, this area is often associated to failure.

This behavior is also reflected in the resolution speedup. Considering for example instance 15757, an optimal solution is found for $s_{max} = \{4, 8, 12, 24\}$ with respective cpu-time of of 514, 7, 299 and 3810 seconds. Simultaneously, the tree width ratio varies from 0.63 for $s_{max} = 4$ to 0.30 for $s_{max} = \{8, 12\}$ and then to about 0.2 for $s_{max} = 24$. The optimal resolution time occurs for $s_{max} = 8$. Several instances follow a similar behavior. Most of solved instances are optimally resolved for a specific $s_{max}$ value . This behavior follows very likely from the compromise between the gains provided by the decomposition and the losses related to a decrease in the freedom of choice during search in dynamic variables ordering in smaller clusters.

## 3 Conclusion

In this paper , we specifically addressed the tagSNP selection problem, a variant of the covering problem coming from bioinformatics. Our WCSP model allows to take into account additional criteria such as dispersion and representativity. The data used in our experimentation is likely the largest data set available in the public domain. With a usual threshold ($\theta = 0.8$), our model provides good results. But the situation quickly become more challenging if less stringent (more artificial) filtering conditions such as $\theta = 0.5$ are used. In the near future, we intend to extend our study to include an integer linear programming model solved using CPLEX.

Beside this, from the methodological point of view, this work underlines the fact that the practical exploitation of tree decompositions to solve structured combinatorial optimization problems is not straightforward. Our experiments show that, even on problems that have a nice visible structure, it is often very profitable and sometimes crucial to restrict the maximum size of the separators of the decomposition used. Theory says that separator size influences the space complexity of structure-based algorithms such as BTD and RDS-BTD but in practice, the improvement in efficiency is mostly explainable by the added freedom in variable ordering allowed by cluster merging, an observation consistent with jegou et al. conclusions [7].

Our current algorithm still leaves areas for improvements. A attractive direction would be to try to design a dedicated heuristic allowing for dynamic adaptation of the maximum separator size of the decomposition. This could be based on the specific structure of the input instance and could be achieved, for a given decomposition, by analyzing treewidth variations as a function of $s_{max}$. Another more challenging direction would be to provide decomposition algorithms aimed at producing small separators in the graph decomposition.

*Acknowledgments* This research has been partly funded by the French *Agence Nationale de la Recherche* (`STALDECOPT` project).

## References

1. A. Choi, N. Zaitlen, B. Han, K. Pipatsrisawat, A. Darwiche, and E. Eskin. Efficient Genome Wide Tagging by Reduction to SAT. In *Proc. of WABI-08*, volume 5251 of *LNCS*, pages 135–147, 2008.
2. B. Cabon, S. de Givry, L. Lobjois, T. Schiex, and J.P. Warners. Radio Link Frequency Assignment. *Constraints*, 4(1):79–89, 1999.
3. C. S. Carlson, M. A. Eberle, M. J. Rieder, Q. Yi, L. Kruglyak, and D. A. Nickerson. Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium. *Am. J. Hum. Genet.*, 74(1):106–120, 2004.
4. M. Cooper and T. Schiex. Arc consistency for soft constraints. *Artificial Intelligence*, 154:199–227, 2004.
5. S. de Givry, M. Zytnicki, F. Heras, and J. Larrosa. Existential arc consistency: Getting closer to full arc consistency in weighted CSPs. In *Proc. of IJCAI-05*, pages 84–89, Edinburgh, Scotland, 2005.
6. J.N. Hirschhorn and M.J. Daly. Genome-wide association studies for common diseases and complex traits. *Nature Reviews Genetics*, 6(2):95–108, 2005.
7. P. Jégou, S. N. Ndiaye, and C. Terrioux. Dynamic management of heuristics for solving structured CSPs. In *Proc. of CP-07*, pages 364–378, Providence, USA, 2007.
8. C. Lecoutre, L. Sais, S. Tabary, and V. Vidal. Last conflict based reasoning. In *Proc. of ECAI-2006*, pages 133–137, Trento, Italy, 2006.
9. Z. S. Qin, S. Gopalakrishnan, and G. R. Abecasis. An efficient comprehensive search algorithm for tagsnp selection using linkage disequilibrium criteria. *Bioinformatics*, 22(2):220–225, 2006.
10. D.J. Rose. Tringulated graphs and the elimination process. *Journal of Mathematical Analysis and its Applications*, 32, 1970.
11. M. Sanchez, D. Allouche, S. de Givry, and T. Schiex. Russian doll search with tree decomposition. In *In Proc. of IJCAI'09*, Pasadena, USA, July 11-17 2009.

12. C. Terrioux and P. Jégou. Bounded backtracking for the valued constraint satisfaction problems. In *Proc. of CP-2003*, pages 709–723, Kinsale, Ireland, 2003.
13. C. Terrioux and P. Jegou. Hybrid backtracking bounded by tree-decomposition of constraint networks. *Artificial Intelligence*, 146(1):43–75, 2003.