A SAT Encoding for the Social Golfer Problem

Inês Lynce
IST/INESC-ID, Technical University of Lisbon
Lisbon, Portugal
ines@sat.inesc-id.pt

Abstract

We introduce a SAT encoding for the social golfer problem. Since 1998, the social golfer problem has become a famous combinatorial problem. It is problem number 10 in CSPLib (http://www.csplib.org/). For a social golfer problem w-p-g, the goal is to schedule a golf tournament during w weeks. Each week, g groups of p players each are formed. No golfer plays in the same group as any other golfer more than once.

1 The Problem

The social golfer problem is derived from a question posted to sci.op-research in May 1998:

> The coordinator of a local golf club has come to you with the following problem. In her club, there are 32 social golfers, each of whom play golf once a week, and always in groups of 4. She would like you to come up with a schedule of play for these golfers, to last as many weeks as possible, such that no golfer plays in the same group as any other golfer on more than one occasion.

In other words, this problem can be described more explicitly by enumerating four constraints which must be satisfied:

1. The golf club has 32 members.
2. Each member plays golf once a week.
3. Golfers always play in groups of 4.
4. No golfer plays in the same group as any other golfer twice.

A solution is said to be optimal when *maximum socialisation* is achieved, i.e. when one golfer plays with as many other golfers as possible. Clearly, since a golfer plays with three new golfers each week, the schedule cannot exceed 10 weeks. This follows from the fact that each golfer plays with three other golfers each week. Since there is a total of 31 other golfers, this means that a golfer runs out of opponents after 31/3 weeks.

week	group 1	group 2	group 3
1	1 2 3	4 5 6	7 8 9
2	1 4 7	2 5 8	3 6 9
3	1 5 9	2 6 7	3 4 8
4	1 6 8	2 4 9	3 5 7

Figure 1: A solution for problem 4-3-3.

For some years, it was not known if a 10 week (and therefore optimal) solution for 32 golfers existed. In 2004, Aguado found a solution using design-theoretic techniques [1]. No constraint programming technique has yet solved this instance, so it remains a valuable benchmark. The best known solution from constraint programming is from Stefano Novello, who found out a 9-week solution by using ECL^{i}PS^e. Hence, the current challenge is to find a 10-week solution.

Even though the social golfer problem was described for 32 golfers playing in groups of 4, it can be easily generalized. An instance to the problem is characterized by a triple w-p-g, where w is the number of weeks, p is the number of players per group and g is the number of groups. The original question therefore is to find a solution to the w-4-8 problem, with w being the maximum, i.e. to find a solution to 10-4-8 (or prove that none exists). For example, Figure 1 gives a solution for the social golfer problem 4-3-3, i.e. for scheduling 9 golfers playing in 3 groups of 3 golfers each for 4 weeks.

The social golfer problem is related with other well-known combinatorial problems. Indeed, this problem is a generalisation of the problem of constructing a round-robin tournament schedule, the main difference being that in the social golfer problem the number of players in a group may be greater than two. Also, the social golfer problem of finding a 7 week schedule for 5 groups of 3 players (5-3-7) is the same as Kirkman's schoolgirl problem.

2 A SAT Encoding

To encode the social golfer problem as a SAT problem we must define a set of variables and a set of constraints (represented by clauses) on the variables.

The set of constraints must guarantee that each golfer plays golf once a week, golfers always play in groups of a given size and no golfer plays in the same group as any other golfer twice.

We have defined SAT variables based on the golfers. Apparently, for a social golfer problem *w-p-g* it should be enough to have $w \times (p \times g) \times g$ variables. The value of each variable would allow us to conclude whether, in a given week, a certain golfer is scheduled to play in a particular group.

However, we have chosen another model. Even though this model has more variables, these variables are quite useful for defining the problem constraints. Instead of $w \times (p \times g) \times g$ variables, this new model has $w \times (p \times g) \times (p \times g)$ variables. When compared with the other model, the difference is that we introduced an additional order relation for golfers within groups. This means that the value of each variable indicates whether golfer i is scheduled to play in group k of week l as the j^{th} player, with $1 \leq i \leq x$, $1 \leq j \leq p$, $1 \leq k \leq g$ and $1 \leq l \leq w$. Although the order of players is irrelevant within groups (as well as the order of groups within weeks and the order of weeks), this model requires most constraints to be at-least-one and at-most-one clauses.

The next step consists in adding clauses to specify that:

- Each golfer plays exactly once per week:
  - At least once per wee.
  - At most once per week.
- Each group in each week has exactly p players:
  - At least one golfer must play as the j^{th} golfer, with $1 \leq j \leq p$.
  - At most one golfer can play as the j^{th} golfer, with $1 \leq j \leq p$.

Let us now consider the social golfer problem *w-p-g*, where the number of golfers is given by $x = p \times g$. Consider GOLFER_{ijkl} to be a variable equivalent to having golfer i playing as the j^{th} player of group k during week l, with $1 \leq i \leq x$, $1 \leq j \leq p$, $1 \leq k \leq g$ and $1 \leq l \leq w$.

Each at-least-one clauses referring to golfers has size $x = p \times g$ and is obtained as simply as follows.

$$\bigwedge_{i=1}^{x} \bigwedge_{l=1}^{w} \bigvee_{j=1}^{p} \bigvee_{k=1}^{g} \text{GOLFER}_{ijkl}$$

The at-most-one clauses referring to golfers are encoded with two sets of binary clauses. The first set of clauses guarantees that each golfer plays at most once in the same group.

$$\bigwedge_{i=1}^{x} \bigwedge_{l=1}^{w} \bigwedge_{j=1}^{p} \bigwedge_{k=1}^{g} \bigwedge_{m=j+1}^{p} \neg\text{GOLFER}_{ijkl} \vee \neg\text{GOLFER}_{imkl}$$

The second set of clauses guarantees that each golfer plays at most once per week.

$$\bigwedge_{i=1}^{x} \bigwedge_{l=1}^{w} \bigwedge_{j=1}^{p} \bigwedge_{k=1}^{g} \bigwedge_{m=k+1}^{g} \bigwedge_{n=j+1}^{p} \neg\text{GOLFER}_{ijkl} \vee \neg\text{GOLFER}_{inml}$$

Let us now consider the clauses referring to groups of golfers. Each at-least-one clause has size x and is obtained as follows.

$$\bigwedge_{l=1}^{w} \bigwedge_{k=1}^{g} \bigwedge_{j=1}^{p} \bigvee_{i=1}^{x} \text{GOLFER}_{ijkl}$$

Finally, the at-most-clauses for groups of golfers are encoded by a set of binary clauses.

$$\bigwedge_{l=1}^{w} \bigwedge_{k=1}^{g} \bigwedge_{j=1}^{p} \bigwedge_{i=1}^{x} \bigwedge_{m=i+1}^{x} \neg\text{GOLFER}_{ijkl} \vee \neg\text{GOLFER}_{imkl}$$

With the set of variables and clauses described above we have encoded all the constraints of the problem, except the one that mentions that "no golfer plays in the same group as any other golfer twice". To guarantee this condition, we introduce a set of auxiliary variables and a *ladder* matrix.

The set of auxiliary variables allows us to know exactly which golfers are scheduled to play in each match. Hence, we must have $x \times g \times w$ additional variables. Clearly, the value of these new variables depends on the value of the variables GOLFER described above. Consider these new variables to be a set of variables denoted as GOLFER'_{ikl}, meaning that golfer i is scheduled to play in group k during week l, with $1 \leq i \leq x$, $1 \leq k \leq g$ and $1 \leq l \leq w$. It is easy to establish an equivalence relation between each variable GOLFER'_{ikl} and the corresponding GOLFER variables. (Each equivalence may be readily converted into a set of clauses.)

$$\text{GOLFER'}_{ikl} \leftrightarrow \bigvee_{j=1}^{p} \text{GOLFER}_{ijkl}$$

These new variables will now be used by the variables in the ladder matrix in such a way that no golfer plays in the same group as any other golfer more than once.

The ladder matrix [2, 4, 5] consists of a set of $(g \times w) \times \binom{x}{2}$ ladder variables (and also a set of ladder clauses). Intuitively, one would say that the value of each variable denotes whether two golfers are scheduled to play together in a given group of a given week. But we can do better. We can guarantee that every two golfers play together at most once.

Consider the ladder variables to be denoted as LADDER_{yz}, with $1 \leq y \leq g \times w$ and $1 \leq z \leq \binom{x}{2}$. A complete assignment of the ladder variables is said to be *valid* if and only if every row is a sequence of zero or more true assignments followed by false assignments.

$$\forall_y \neg \exists_z \text{LADDER}_{yz} = \text{FALSE} \wedge \text{LADDER}_{yz+1} = \text{TRUE}$$

The behavior of the ladder matrix can be used to guarantee that no two golfers play more than once in the same group. Actually, having an adjacent pair of variables with values TRUE and FALSE identifies precisely in which group of which week two golfers played together.

Whenever a ladder variable is satisfied, there is a set of adjacent variables that must be satisfied. This can be achieved by unit propagation adding the following set of clauses.

$$\bigwedge_{y=1}^{g \times w} \bigwedge_{z=1}^{\binom{x}{2}-1} \neg Ladder_{yz+1} \vee Ladder_{yz}$$

Finally, the variables in the ladder matrix must be related with the auxiliary variables described above (denoted as GOLFER'). If one of these variables is satisfied, meaning that one golfer is scheduled to play in a specific group, then the corresponding ladder variables are satisfied. Obviously, the ladder variables to be satisfied depend on the golfers that are also scheduled to play in the same group.

Appendix: Symmetry Breaking

This social golfer problem is highly symmetric, exhibiting the following symmetries:

- Golfers are interchangeable. That is, the names of the 32 golfers are insignificant.
- Golfers within a group are interchangeable. Order has no significance for groups of golfers.
- Groups within a week are interchangeable. Again, order has no significance when considering groups within a week.
- Weeks are interchangeable. There are no order constraints with respect to weeks.

For example, considering again the solution given in Figure 1, one may assume that symmetries have been eliminated: this explains why golfers are ordered within groups, groups are ordered within weeks with respect to the first player and weeks are ordered with respect to the second player of the first group.

After establishing the model described above, we have considered adding clauses to our SAT encoding for breaking symmetries. However, experimental results given in [3] indicate that symmetry breaking does not speed-up SAT solvers in our encoding. Hence, we decided not to include symmetry breaking clauses in the benchmarks we submitted to the competition.

References

[1] Alejandro Aguado. A 10 days solution to the social golfer problem, 2004. Manuscript.

[2] Carlos Ansótegui and Felip Manyá. Mapping problems with finite-domain variables into problems with boolean variables. In *SAT'04*, 2004.

[3] Ian P. Gent and Inês Lynce. SAT encodings for combinatorial problems, 2005. Submitted.

[4] Ian P. Gent and Peter Nightingale. A new encoding of alldifferent into sat. In *3rd International Workshop on Modelling and Reformulating Constraint Satisfaction Problems, CP'04*, 2004.

[5] Ian P. Gent and Patrick Prosser. Sat encodings of the stable marriage problem with ties and incomplete lists. In *SAT'02*, 2002.