Scheduling a Bridge Club by Tabu Search

Luis B. Morales
IIMAS, Universidad Nacional Autónoma de México
Apdo. Postal 70-221, México, DF, 04510, Mexico
lbm@servidor.unam.mx

Introduction A bridge club scheduling problem was presented in [1], as follows: A bridge club consists of 12 couples. At each meeting, the club is divided into 3 groups of 4 couples. Each of the 4 couples then plays against the remaining 3 couples in its group, which requires 6 games per group. Thus 18 games are played at each meeting, of which 8 occur per year, resulting in a total of 144 games. The club needs to prepare a schedule for the year such that the number of times any couple plays any other couple should be the same for all couples. If no such schedule exists, then the club needs to prepare a schedule for the year such that the number of times any couple plays any other couple is most nearly the same for all couples.

The authors of [1] formulated this scheduling problem as a discrete optimization problem, and used four techniques of combinatorial optimization: greedy algorithm, branch and bound, steepest descent and simulated annealing (see e.g.,[3]) to seek an optimal schedule. None of their methods produced an optimal schedule.

An optimal solution In the course of the year's eight meetings each couple competes against 24 couples (teams). Since a couple has only 11 different possible opponents, an "optimal" schedule would have each couple playing 9 other couples twice and 2 other couples three times. However, there is no a priori guarantee that such an "optimal" schedule exists.

Define the competition matrix of a schedule as the matrix (C_{ij}), where C_{ij} is the number of times team i competes against team j in that schedule.

This matrix is symmetric ($C_{ij} = C_{ji}$), with 0's on the diagonal, so only the 66 elements above (or below) the main diagonal need to be examined. If an optimal schedule exists, then each row of C would consist of nine 2's and two 3's (with 0's on the diagonal).

A *cost function* $f(x)$ of a particular schedule x can be computed by summing the square of the difference between the ideal number of times two teams should meet, namely 2, and each of the entries above the main diagonal of the competition matrix. That is,

$$f(x) = \sum_{i<j}^{12} (C_{ij} - 2)^2.$$

Clearly, $f(x)$ decreases when the competition matrix C_{ij} has more entries that are closer in value to 2; the minimum value for $f(x)$ is thus 12. Using a simple tabu search method (TS) we will find an optimal schedule of cost 12 (see [2] for more details on TS).

Tabu Search TS is a general search procedure for solving combinatorial optimization problems of the general type

$$\text{minimize } f(x) \quad \text{subject to } x \in X,$$

where f is a cost function and X is a set of feasible solutions. The procedure has its roots in the intelligent problem solving approach, found in artificial intelligence. (By contrast, simulated annealing is based in the second law of thermodynamics, from physics.)

The TS method is an iterative process; it starts from an initial feasible solution and tries to reach a global minimum by moving step by step. To accomplish this, we must define a set M of simple modifications that can be applied to a given solution in order to move to another. These modifications are called *moves*. The notation $x' = m(x)$, $m \in M$ indicates that m transforms x into x'. This leads us to the definition of a *neighborhood*, an ingredient common to most heuristic and algorithmic procedures for optimization. For each feasible solution x, the neighborhood $N(x)$ is the set of all feasible solutions directly reachable from x by a simple move m in M. At each step of the iterative process, we generate a subset V^* of $N(x)$, and we move from x to the best solution x^* in V^*, whether or not $f(x^*)$ is better than $f(x)$. (If $N(x)$ is not large, it is possible to take $V^* = N(x)$.)

Up to this point, the algorithm is close to a local improvement technique, except that we may move from x to a worse solution x^*, and thus may escape from local minima of f. (This situation occurs in simulated annealing, where a non-improving move may be accepted, with an exponential probability distribution that decreases as the number of iterations grows.) To prevent cycling, any move that returns to any local optimum recently visited is forbidden. This is accomplished in a *short-term memory* framework by storing the forbidden (tabu) move in a set T, called a *tabu list*, of length $|T| = t$ (the length may be fixed or variable). A move remains forbidden during t iterations, so the tabu list can be represented by a queue: At each iteration, the opposite move from x to x^* is added at the end of T, while the oldest is removed from T.

Unfortunately, the tabu list may forbid certain interesting moves, such as moves that lead to a better solution than the best one found so far. An *aspiration criterion* is introduced to cancel the tabu status of a move when this move is judged useful.

Stopping rules must also be defined; in many cases (including the bridge club problem) a lower bound f^* of f is known in advance. As soon as we are close enough or we have reached this bound, we may interrupt the algorithm. In general, however, f^* is not available with sufficient accuracy, so a fixed maximum number of iterations is given.

Adaptation of tabu search In our approach a feasible solution (or schedule) is an arrangement of eight columns into which the 12 couples are assigned to the three groups A, B and C; each group contains four couples. By the preceding definition, x' is a neighboring solution of x if x and x' are identical for every meeting but one, and in the exceptional meeting, exactly two teams switch groups when moving from x to x'. A move is entirely defined by one meeting and two teams in that meeting, which belong to different groups. Each feasible solution x has $8 \times 3 \times 4 \times 4 = 384$ neighbors, because there are 8 different meetings in which a neighbor may differ, and in each meeting, there are 3 ways to choose the groups involved in the switch and 4 ways to choose a team from each of the groups.

An initial feasible solution is generated as follows: For each column of the schedule, we randomly split the 12 couples into the groups A, B, and C, with four couples each.

To construct the tabu list, we forbid the exchange of any two teams that were exchanged between two groups at a meeting during the preceding t iterations, where t is the tabu size. Formally, the tabu list consists of triples (i, j, k), where i and j are teams that cannot be exchanged in meeting k. The value of t is crucial for the performance of TS, as we will later see.

For the problem at hand we define the aspiration criterion as a function $A(z)$ defined for every value z of the cost function. The criterion allows a tabu move to be selected if it leads to a solution better than the best found so far. This means that for any solution x, $A(f(x)) = f(x^o)$ where x^o is the best solution found so far; the tabu status of a move from x to x' may be dropped if $f(x') < A(f(x))$.

Heuristic techniques, including TS, do not always guarantee reaching an optimal solution. So the process is stopped if the given maximum number of iterations has been reached. On the other hand, we must stop the process if the cost function f has reached the optimal value, in this case 12.

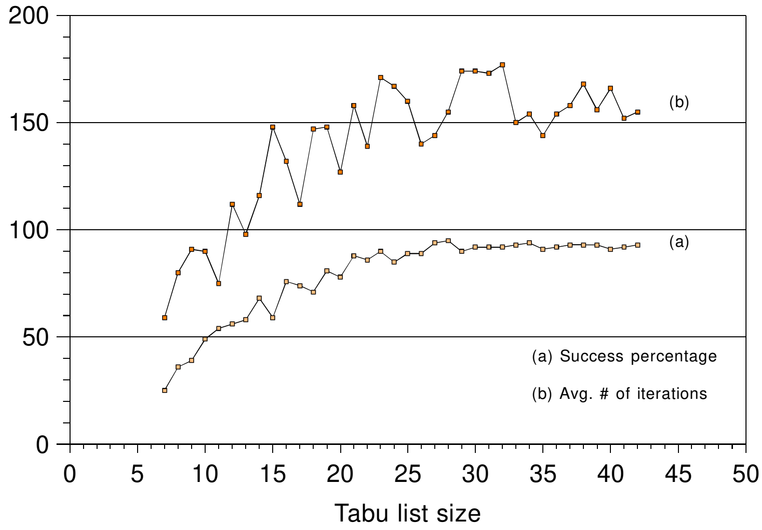

Figure 1. (a) A plot of the number of times that TS found an optimal solution out of 100 runs vs tabu sizes. (b) A plot of the mean number of iterations to obtain an optimal solution vs tabu size.

Results In order to assess good values for the tabu list size t, we performed TS 100 times with a random initial schedule and a maximum of 500 iterations, for each t between 7 and 42. In Figure 1 we have plotted the (a) number of times an optimal solution was found out of 100 (success percentage), and (b) the mean number of iterations needed to find it for each value of t in $[7, 42]$. Note that in the interval $[7, 29]$ both the success percentage of TS and the mean number of iterations (time complexity to find optimality) increase when the tabu list size increases. However, for tabu size values between 30 and 42, the success percentage seems to stay constant, and the time to solve optimality tends to grow. Hence, the best tabu sizes seem to be integers somewhere between 26 and 29. With these sizes TS always found the solution of cost 14 reported in [1]. For a small tabu size ($t < 16$), cycling was detected with a high probability. But if the tabu size was too large, the number of iterations to find the optimal schedule became larger too.

Also in our experiments we have seen that for a fixed tabu list size, the number of iterations needed to find an optimal solution strongly depends on the initial schedule. For example, a schedule of cost 12 was found in 9 iterations with one initial solution, while about 400 iterations were required to reach an optimal solution using a different initial schedule.

At each step of the TS process, we examined the entire neighborhood; that is $V^* = N(x)$ for each current solution x. This method needs more calculation time than the partial inspection of the set $N(x)$, but the results are better. However, the complete examination becomes very expensive as the problem size increases. For large problems, V^* can be restructured as follows: We examine the neighbors and take the first that improves the current solution x. If no move improves x, or if every improving move is tabu, then we examine $N(x)$. Another approach consists in randomly generating V^* with a fixed number of neighbors of x.

An optimal schedule obtained by TS and its competition matrix are given in Figure 2.

Schedule
(entries are group label assigned to team)

Team #	Meeting # 1	2	3	4	5	6	7	8
1	B	C	C	B	B	A	B	B
2	A	B	C	C	C	B	B	A
3	C	A	C	B	A	B	C	C
4	A	A	A	B	C	C	A	B
5	B	A	A	A	C	A	C	A
6	C	B	B	B	B	A	A	A
7	C	C	B	C	C	C	C	B
8	A	A	B	C	A	A	B	C
9	B	C	A	C	B	B	A	C
10	B	B	B	A	A	B	A	B
11	A	B	C	A	B	C	C	C
12	C	C	A	A	A	C	B	A

Competition Matrix
(entries are # of times team plays opponent)

Team #	Opponent # 1	2	3	4	5	6	7	8	9	10	11	12
1	0	2	2	2	2	3	2	2	3	2	2	2
2	2	0	2	2	2	2	2	3	2	2	3	2
3	2	2	0	2	2	2	2	3	2	2	3	2
4	2	2	2	0	3	2	3	2	2	2	2	2
5	2	2	2	3	0	2	2	2	2	2	2	3
6	3	2	2	2	2	0	2	2	2	3	2	2
7	2	2	2	3	2	2	0	2	2	2	2	3
8	2	3	3	2	2	2	2	0	2	2	2	2
9	3	2	2	2	2	2	2	2	0	3	2	2
10	2	2	2	2	2	3	2	2	3	0	2	2
11	2	3	3	2	2	2	2	2	2	2	0	2
12	2	2	2	2	3	2	3	2	2	2	2	0

Figure 2. An optimal schedule and its competition matrix.

Conclusion The tabu list size is an important for the performance of the tabu search heuristic method. We have analyzed the performance of TS with different tabu list sizes; likewise, we found the best range of adequate tabu list sizes for our implementation of TS. For some of these tabu sizes our procedure obtained, with high probability (up to 95%), an optimal solution for the scheduling problem.

Additional Note. After this paper was submitted for publication a paper by Simpson was submitted entitled Scheduling a bridge club using a genetic algorithm (this issue). The author solve the same scheduling problem, but using a genetic algorithm.

References

[1] B. S. Elenbogen, and B. R. Maxim, Scheduling a bridge club (a case study in discrete optimization), this magazine 65(1992), 18-26.

[2] F. Glover, Tabu Search Part I, *ORSA Journal on Computing* 1(1989), 190-206.

[3] S. Kirkpatrick, C. D. Gelatt Jr., and M. Vecchi, Optimization by simulated annealing, *Science* 220(1983), 671-680.