Open access • Journal Article • DOI:10.1287/INTE.1080.0363

Showcase Scheduling at Fred Astaire East Side Dance Studio — Source link

Miguel A. Lejeune, Nevena Yakova

Institutions: Drexel University

Published on: 01 May 2008 - Interfaces (INFORMS)

Topics: Dance, Schedule and Studio

SHOWCASE SCHEDULING AT FRED ASTAIRE EAST SIDE DANCE STUDIO

M.A. Lejeune*, N. Yakova**

* LeBow College of Business, Drexel University, mlejeune@drexel.edu
** Fred Astaire East Side Dance Studio, nyakova@yahoo.com

Abstract:

The ballroom dancing showcases at Fred Astaire East Side Dance Studio in Manhattan are held at least twice a year and provide the students with an environment for socializing, practice, and improvement. The most important part of a showcase organization is the construction of the dance presentations timetable, and, with the number of participants increasing each year, this task (usually performed manually) requires a great deal of time and effort on the part of the management. That is why, for the Summer 2007 showcase, the management of the studio expressed interest in using a computer-based tool that would significantly speed up the process of schedule construction and deliver a much more coherent and efficient showcase program. The tool we developed is rooted in coding of algorithms and integer programming optimization models and is very user-friendly since it is spreadsheet-based and runs in Excel, while taking into consideration the various constraints and requirements posed by the management, instructors, and students of the dance studio. The scheduling tool was successfully implemented for the generation of the timetable of the above-mentioned showcase; the management spent considerably less time preparing this schedule as compared to the previous practice of manual program construction. Therefore more time is now afforded to the other important activities of the studio such as teacher training, dance instruction, sales, marketing, and communication with clients.

Key words: Programming: Integer: Applications; Recreation – Sports: Dance; Sports Scheduling

---

One seldom sees the expressions "ballroom dancing" and "operations research" in the same sentence. However, operations research techniques prove to be applicable in the context of scheduling and organization of ballroom dance competitions and showcases. Fred Astaire East Side Dance Studio in Manhattan, NY, has recently put into test the use of OR techniques to construct the timetable of their dance showcases. The coding of the algorithms and the integer programming optimization models in Visual Basic provide a very user-friendly spreadsheet-based decision-support tool that can generate the schedule of a showcase while accounting for diverse constraints rooted in clients' and instructors' preferences among others.

Dance instruction is one of the fastest growing sectors in the US with the number of social and competitive dancers of all ages continuously growing. The industry is gaining momentum due to increased exposure in the popular culture, through TV shows and motion pictures. Besides as a form of physical exercise, ballroom dancing is viewed as a means and environment for socializing. Additionally, due to its widespread popularity, ballroom dancing, in its competitive and sportier version – DanceSport - is being debated for possible inclusion in the Olympic Games.

Fred Astaire East Side Dance Studio is a part of the franchise network of Fred Astaire Dance Studios, Inc., headquartered in Longmeadow, MA, and functioning in the United States and Canada. The franchise was named after the famous Fred Astaire, who was also one of the co-founders. The original Fred Astaire Dance Studio was founded in New York in 1947 and became a franchise in 1950. Currently, there are over 110 Fred Astaire studios in North America.

The studio offers instruction in all kinds of social dance (*e.g.,* ballroom, country western, nightclub dancing etc.) through private, group, and practice lessons. In addition, as a regular part of their annual calendar, every Fred Astaire studio organizes different kinds of social events for their students – parties, showcases, and competitions – where the students can demonstrate their performance skills and technique, and receive feedback on their progress. Particularly, in the seasonal showcases offered by the Fred Astaire studios (usually at least twice a year), students are paired with their teachers to demonstrate their achievements on the dance floor in a noncompetitive environment, in front of friends and family who are invited as guests. The problem at hand was presented by the management of Fred Astaire East Side Dance Studio and concerns the time-scheduling of those annual showcases.

A showcase is usually at least a half-day event that consists of dance demonstrations structured in rounds called “Heats” that last around 1min 15sec. Each Heat represents one kind of dance (*e.g.,* Tango, Waltz, Foxtrot etc) which is performed simultaneously on the dance floor by several student-teacher couples. Students participate in a showcase upon payment of fee per entry; an entry is the performing of one dance in a traditional Heat. The total number of entries in a showcase can reach 1000. In addition, students are given the opportunity to present a “Solo” for which the floor is reserved only for that particular student and his/her teacher. Unlike the Heats, the Solo dances usually are of various and longer duration (sometimes up to 3min), and incorporate custom choreography that reflects each student’s ability and interests. Hence, the fee for a Solo entry is higher than the fee for a Heat entry.

Considering the above descriptions, and the requirements and constraints discussed in the next part of the paper, the management of Fred Astaire East Side Dance Studio (and, almost certainly, the management of many other dance studios) is presented with the extremely tedious and time-consuming task of constructing the showcase schedules manually since no suitable software is available and/or affordable. Indeed, competition scheduling software used for the planning of big ballroom dance competitions exists; however, these competitions consist of many more and different parameters, among which age grouping and grouping according to class of dance expertise. Therefore, the software used for their scheduling proves to be too cumbersome if applied to the scheduling of the studios’ showcases, and the studio managers are reluctant to invest in something costly that will not be able to address their particular needs. As a result, the managers are forced to spend weeks manually constructing and tweaking the schedules. Even after the construction of the pencil-based schedule, the managers must check again whether all dances have been scheduled, whether each customer has been paired with the asked teacher, and whether all the other constraints have been adequately accounted for. The task of the managers is additionally exacerbated in the cases when students decide (sometimes days before the showcase) to change their initial registration, either by canceling or adding entries. Although such changes are only allowed up to a reasonable deadline before the showcase, they would still require the managers to spend more time rewriting the schedule in order to comply with the general constraints mentioned earlier. To decrease the time spent on the schedule construction, Fred Astaire East Side Dance Studio contacted us with the request to automate this process.

Inputs and Requirements

The objective here is to construct a showcase schedule that satisfies the requirements of the customers. The problem is related, to some extent, to the sports scheduling (see Dinitz et al., 2006, Trick, 2007 for a review) discipline and to multi-period assignment problems (Gilbert, Hofstra, 1987) involving the construction of schedules for employees (Ernst et al., 2004). The problem at hand though distinguishes itself from the latter area by the high emphasis placed on the satisfaction of the desiderata of the students, which, for paid workers, is of lower importance. In this paper, we propose both a greedy heuristic and integer optimization problems to construct the showcase schedule.

Each showcase features up to 30 different types of dances, and each student can register for an unlimited number of dances. Additionally, each student indicates the type of dance she wants to register for, which teacher she will dance with, the number of times she wants to dance with that teacher, and the number of times she wants to dance the same type of dance with the same teacher. Therefore, in order to differentiate between these perfect duplicate cases (see rows 6, 7, 8 in Table 1) where the same student-teacher pair performs the same type of dance, we introduce an indicator w (see Appendix). Thus the inputs for the construction of the schedule are introduced in an Excel file and provide (i) the entries which are 4-tuple of information characterized by the name of the student, the name of the teacher, the type of dance, and the indicator w (Table 1); (ii) the availability of the students (Table 2); and (iii) the availability of the teachers (Table 2).

Entries			
Names of Students	**Names of Teachers**	**Types of Dance**	***w***
S1	T1	Merengue	1
S2	T2	American Cha-cha	1
S3	T3	East Coast Swing	1
S4	T4	American Tango	1
S5	T5	Quickstep	1
S6	T5	American Foxtrot	1
S6	T5	American Foxtrot	2
S6	T5	American Foxtrot	3
S6	T6	Salsa	1
S7	T7	American Cha-cha	1

Table 1: Entries: The first and second columns respectively indicate the names of the students and of the teachers. The third column reports the type of dance and the last one is the value taken by the indicator variable differentiating duplicate entries when a student has registered to dance several times the same type of dance with the same teacher.

Students	Non-Availability		Teachers	Non-Availability	
	From	To		From	To
S1	10:30 AM	12:30 PM	T1		
S2	11:00 AM	2:00 PM	T2		
S3			T3	1:00 PM	1:30 PM
S4			T4		
S5			T5		

Table 2: Availability of Students and Teachers: The first and fourth columns report the names of the students and teachers while columns 2 and 3 (respectively, 5 and 6) indicate the times (starting and ending) during which the students (respectively, teachers) are not available.

Each showcase requires the construction of a complex schedule that orders the Heats and assigns the students and teachers according to several rigid constraints:

1) A single type of dance (*e.g*., Tango, Viennese Waltz) must be assigned to every Heat.
2) Each registered dance must be assigned to a Heat which features the type of dance of the registered dance.
3) Two consecutive Heats cannot present the same type of dance (*i.e*., a Tango cannot be followed by a Tango).
4) No student and/or teacher should be listed twice in the same Heat.
5) Teachers might have a limited availability during the showcase.
6) A student can dance the same dance as many times as she wants (as long as it is not in a row, as mentioned in 3).

7) A student can dance with the same teacher as many times as she wants.

8) A student can dance the same type of dance with the same teacher as many times as she wants.

9) A student can dance with as many teachers as she wants.

10) The number of the Heats is not limited.

11) Except scheduled breaks (*e.g.*, meals, collations), there can't be any interruption (*i.e.*, Heat without any scheduled dance followed by other Heats with scheduled dances) in the program.

12) At most seven student-teacher pairs can dance at a time (*i.e.*, within one Heat).

Additionally, apart from the rigid constraints, the management requested for several secondary issues to be addressed if possible. Specifically: (i) to minimize the duration of the showcase (*i.e.*, minimize the total number of Heats); (ii) to accommodate students who ask to participate only at specific time during the showcase; (iii) to minimize the number of Heats where only one couple is dancing (this situation is almost inevitable due to the high student-teacher ratio but needs to be minimized because of the mentioned earlier price difference between a Heat entry and a Solo entry); and (iv) to keep the students involved throughout the whole showcase and avoid placing the same groups of couples in consecutive Heats (*i.e.* the participants need to be shuffled throughout the whole showcase). Note that the rigid requirements that prevent from scheduling two consecutive Heats featuring the same type of dance and from having any interruption could lead to infeasibility for some specific showcases (*e.g.*, for example a showcase in which the number of Heats for a type of dance is larger than one plus half the total number of Heats in the showcase). In such situation, we would consider the two consecutive Heats with same type if dance as a soft constraint.

Finally, as far as the Solo entries are concerned, we were advised that they will be incorporated in the final program as a separate section somewhere in the middle of the Heats program. The Solos are each of variable duration and music, they are rather limited in number, and they often entail change of costumes. Therefore, the studio decided to schedule them one after another, as one block, thus avoiding complications in the construction of the overall program (since Heats are of equal duration). Incidentally, this would give most of the participants an opportunity for a longer break and the guests would enjoy a mini-program of more artistic demonstrations. Thus, when proceeding with the development of our models, only the scheduling of the regular Heats will be taken into consideration.

Construction of showcase schedule

The problem posed here is essentially one of feasibility. The main goal is to find a schedule that meets all of the conditions imposed on it. The dance studio's Summer 2007 showcase, which we used to test-drive our solution, required the scheduling of 583 Heat entries featuring 19 types of dances and 18 Solo entries. The participants consisted of 28 students and 8 teachers.

Prior to constructing a schedule that satisfies the constraints listed above, and in particular that minimizes the number of one-couple Heats, we develop an algorithm that provides a close estimate on the minimal total number of Heats to be incorporated in the program. Second, we formulate an integer programming problem (MODEL I) whose optimal solution gives the true optimal (*i.e.*, minimal) number of Heats in the program. The computed estimate (lower bound) or true minimal value of the number of Heats will then be used as an input in an heuristic and in an integer model (MODEL II) that accounts for the desire of the studio to minimize the number of one-couple Heats. MODEL I and MODEL II are solved with the CPLEX 10.2 integer solver. The complete formulation of the two models is given in the Appendix.

Minimum Number of Heats in Showcase

We first discuss the algorithm that computes a lower bound N on the minimal number of Heats to be included in the schedule. A first estimate $\tilde{N}$ of N is computed by taking into account the total number of entries $\sum_{i \in I} N'(d,i)$ for each type d of dance and the maximum number (7) of couples that can dance during a Heat:

$$\tilde{N}(d) = \left\lceil \sum_{i \in I} \frac{N'(d,i)}{7} \right\rceil = \left\lceil \sum_{j \in J} \frac{N''(d,j)}{7} \right\rceil \quad \text{and} \quad \tilde{N} = \sum_{d \in D} \tilde{N}(d) \ .$$

The expression $\lceil a \rceil$ refers to the smallest integer number at least equal to *a*, $N'(d,i)$ is the number of dances of type *d* for which student *i* has registered, $N''(d,j)$ is the number of dances of type *d* that teacher *j* has been chosen to dance, and *I* and *J* are respectively the set of students and teachers participating in the showcase.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
American Viennese Waltz	0	3	14	0	4	0	0	0	0	0
American Foxtrot	3	3	10	3	6	7	5	0	0	5
East Coast Swing	0	5	9	0	4	7	5	0	5	0
Salsa	0	0	9	0	2	4	5	2	0	0
American Tango	3	4	0	3	2	5	5	0	0	5
American Cha-cha	2	4	9	3	7	7	5	3	5	5
West Coast Swing	0	0	0	0	0	5	0	0	0	0
American Rumba	0	2	9	3	6	7	5	2	5	5
American Samba	2	4	0	0	2	5	0	0	5	0
Argentine Tango	0	0	5	0	4	2	2	3	0	5

Table 3: The cells in the above table report, for every type of dance, the minimal number of Heats in which each student will participate. For example, N'(S3,Salsa) = 9 means that the student S3 will participate in 9 Heats reserved for Salsa.

Proceeding along that line provides a valid lower bound on the total number of Heats to be scheduled; clearly, denoting by *r* the true minimal number of Heats to be included, we have $r \geq \tilde{N}$. This bound, however, does not account for the requirements specific to each student and teacher. In what follows, we first describe how, reasoning at a more granular level, we account for such specific requirements; we then explain why this approach is particularly relevant for the problem at hand, and, finally, illustrate with an example the efficiency of the approach and how it can sharpen the bound on the total number of Heats. We proceed as follows:

- computation, for each type *d* of dance, of the minimal number of Heats (Table 4)
    - *N'(d)* when only the students' requirements are accounted for: $N'(d) = \max_i N'(d,i)$;
    - *N''(d,j)* when only the teachers' requirements are accounted for: $N''(d) = \max_j N''(d,j)$;
- computation of
    - the minimal number $\bar{N}(d)$ of Heats per type of dance: $\bar{N}(d) = \max[N'(d), N''(d)]$, and
    - the minimum number $\bar{N}$ of Heats: $\bar{N} = \sum_{d \in D} \bar{N}(d)$,

when the specific requirements of students and teachers are considered.

A few comments are in order. First, $\bar{N}$ constitutes a lower bound on the true minimal number r of Heats to be included: $r \geq \bar{N}$. Second, we note that the estimate $\bar{N}$ takes into account the requirements for the teachers and for the students on an independent basis and does not integrate their joint requirements or their possible limited time availability. Third, for our particular Summer 2007 showcase schedule, we observe that for each of the 19 types of dance, $\bar{N}(d) > \tilde{N}(d)$ (see Table 4, columns 5 and 6). So, obviously, the taking into account of the requirements specific to students and teachers contributes to significantly sharpen the lower bound on the total number of Heats to be included in the showcase. Since, for other problems or showcases, we could very well have the case that $\bar{N}(d) \leq \tilde{N}(d)$, we define the lower bound N on the total number of dances as:

$$N = \sum_{d \in D} \hat{N}_d \qquad \text{with} \qquad \hat{N}_d = \max\left(\bar{N}_d, \tilde{N}_d\right) .$$

The main reason for which $\bar{N}$ is sharper is twofold. First, some students register to dance many times the same type of dance. Table 4 shows that there were 32 American Viennese Waltz (d=AVW) entries for the Summer 2007 showcase. We also notice that one of the students had registered to dance an American Viennese Waltz 14 times. Consequently, $\tilde{N}(AVW) = 5 < \bar{N}(AVW) = 14$, which indicates that at least 14 American Viennese Waltz Heats must be included in the program, and which shows that the bound $\tilde{N}(AVW)$ is looser than $\bar{N}(AVW)$. Second, some teachers appear to be "specialized" in some types of dances and, therefore, are very demanded for those. We observe (Table 4) that a teacher (T6) has been selected to dance Salsa (d=S) 21 times among the 40 Salsa entries. This implies that: $\tilde{N}(S) = 6 < \bar{N}(S) = 21$, which again shows that a much tighter bound can be obtained by considering individual requirements.

	Total Number of Entries	Minimum Number of Heats Students' Requirements	Minimum Number of Heats Teachers' Requirements	Minimum Number of Heats (Number of Entries)	Minimum Number of Heats (Teachers and students' spoecific requirements)
American Viennese Waltz	32	14	9	5	14
American Foxtrot	64	10	24	10	24
East Coast Swing	59	9	19	9	19
Salsa	40	9	21	6	21
American Tango	50	5	19	8	19
American Cha-cha	71	9	24	11	24
West Coast Swing	7	5	5	1	5
American Rumba	74	9	25	11	25
American Samba	21	5	10	3	10
Argentine Tango	50	5	25	8	25

Table 4: Lower bound on the minimal number of Heats. The second column shows the total number of entries per type of dance (*i.e.*, for 10 of the 19 dance types). The minimal number of Heats that must be scheduled if one takes into account the specific requirements for the students (respectively, the teachers) is given in the third (respectively, fourth) column. The fifth column gives the minimal number $\bar{N}(d)$ of Heats for each type of dance *d* if one takes into account the maximum number of students (7) that can dance in a Heat and the total number of entries for each dance type. The last column gives the minimal number $\tilde{N}(d)$ of Heats for each type of dance *d* if one takes into account the specific requirements of both the students and teachers.

We now turn to the formulation of an integer programming optimization model (MODEL I) which provides (when it is solved to optimality) the true minimal number *r* of Heats that must be included in the showcase when all the rigid constraints described above are satisfied and the time availability of students and teachers is considered. The objective function of MODEL I minimizes the duration of the showcase (*i.e.*, minimization of the total number of Heats) and is therefore linear. The rigid requirements are formulated as linear constraints involving binary decision variables which respectively represent whether a registered dance is scheduled to take place during a certain Heat *t*, and whether a certain Heat *t* features a certain type of dance *d*.

The computation of the lower bound (see the algorithm above) on the minimal number of Heats is extremely fast, but, as mentioned earlier, it provides only an estimate on the true minimal number of Heats. We solved MODEL I to optimality in order to: (i) evaluate the quality of the bound provided by the algorithm: for the Summer 2007 schedule, the algorithm generates a lower bound *N* which is equal to the true minimal number *r=240* of Heats that the showcase must feature to accommodate all the requirements; and (ii) benchmark the time needed to solve MODEL I to optimality with the time needed to derive the algorithm-based lower bound; unsurprisingly, the solution time for the integer model is several orders of magnitude larger than the time needed for the algorithm-based solution.

For the construction of the Summer 2007 showcase, the use of the algorithm to compute the minimal number of Heats is an obvious choice. However, for the construction of other showcases and/or for the handling of other types of constraints, MODEL I could also be considered.

Showcase Schedule

We now develop an heuristic method for the construction of a schedule that contains a minimum number of one-couple Heats while addressing all the rigid and soft requirements described above. The heuristic contains two main steps. The first one assigns a type of dance to each Heat in such a way that two consecutive Heats do not feature the same dance. The second step of the heuristic considers the entries defined by the student, the teacher, the type of dance, and the availability of both the student and the teacher, one at a time, and allocates them to Heats. Within the heuristic, the assignment of a registered dance to a Heat is performed in two rounds. In the first round, the heuristic considers the first Heat in the schedule and verifies if the six conditions below are satisfied:

1) the Heat's type of dance corresponds to the considered registered dance;
2) the student is not yet scheduled to dance in this Heat;
3) the teacher is not yet scheduled to dance in this Heat;
4) the student is available at the time at which the Heat is scheduled;
5) the teacher is available at the time at which the Heat is scheduled.
6) the number of entries assigned to the considered Heat is smaller than or equal to 1.

If those six conditions are verified, the registered dance is assigned to this Heat and the heuristic turns to the next registered dance on the list. Otherwise, as soon as one of the above six conditions is not met, the heuristic looks at the next Heat in the schedule and checks whether the registered dance can be assigned to it. If a registered dance could not be assigned to any of the planned Heats, it is placed on a *residual* list. Entries placed on the residual list are then dealt with in the second assignment round in which a registered dance is assigned to a given Heat if the first five conditions described above are met. The heuristic is coded in Visual Basic as a Macro that can be activated through the single click of a button.

Although the heuristic aims at minimizing the number of one-couple Heats, it does not guarantee that the true minimum will necessarily be reached. Therefore, we evaluate the quality of the solution provided with the heuristic with respect to a lower bound on the minimal number *I(d)* of one-couple Heats. Such a bound can be computed for each type of dance by comparing the minimal number *N(d)* of Heats per type of dance and the total number *T(d)* of entries per type (*d*) of dance. More precisely, if the total number *T(d)* of entries per type of dance is larger than or equal to twice the minimal number *N(d)* of Heats, then the minimal number *I(d)* of Heats with one student-teacher pair is equal to 0. Otherwise, if the total number *T(d)* of entries per type of dance is strictly smaller than twice the minimal number *N(d)* of Heats, then the minimal number *I(d)* of Heats with one student-teacher

pair is positive:

$$T(d) \geq 2N(d) \rightarrow I(d) = 0$$
$$T(d) < 2N(d) \rightarrow I(d) = 2N(d) - T(d)$$

	Total Number of Registered Dances	Overall Minimum Number of Heats	Minimum Number of Heats with One Scheduled Student-Teacher Pair
American Viennese Waltz	32	14	0
American Foxtrot	64	24	0
East Coast Swing	59	19	0
Salsa	40	21	2
American Tango	50	19	0
American Cha-cha	71	24	0
West Coast Swing	7	5	3
American Rumba	74	25	0
American Samba	21	10	0
Argentine Tango	29	25	21

Table 5: Lower bound on the minimal number of one-couple Heats. The second column gives the minimal number of entries for every type of dance. The third column provides the bound on the minimal number of Heats per type of dance. The fourth column gives, for each type of dance, the minimal number of one-couple Heats.

We finally developed a second integer optimization model (MODEL II) which:

- minimizes the number of one-couple Heats. This step requires: (i) the introduction of a constraint that enforces that the schedule contains a minimum total number r (whose value is known through the solution of MODEL I) of Heats; (ii) the introduction of binary decision variables which indicate whether a given Heat has exactly one couple scheduled to dance; (iii) the introduction of constraints which force the above binary variables to take value 1 if a given Heat has strictly more than one couple scheduled to dance;
- takes into account the need to keep the students involved during the entirety of the showcase by avoiding the concentration of their entries in one short period. We subdivide the showcase into four main periods (quarters) and require for each student to dance during each of the four quarters, provided that they are available to do so. This requires the introduction of additional binary decision variables for each student. Those variables indicate whether the considered student has at least one dance scheduled in a given quarter.

We used and applied both the heuristic and MODEL II to construct the showcase schedule, and we benchmarked their solution time and the quality of their solution. The solution process for MODEL II turns out to be much more time-consuming than the application of the heuristic approach (see Table 7 in Appendix). Based on that, the studio showed a clear preference for the heuristic, even if its application sometimes left a few entries unassigned (these were incorporated manually in the final schedule). The schedule constructed with the heuristic approach depends on the order in which entries are considered. We considered ten different orderings of the entries and for three of these orderings, all entries were assigned. Other factors that played a role in the choice made by the dance studio were the relative easiness and user-friendliness of the Excel-based process incorporating the heuristic approach, and the fact that the approach based on the integer model would have required the purchase of and the skill to use an optimization solver.

Comprehensible Individual and Showcase Schedule - User-Friendliness of Process

Once the manager has filled in the two input data worksheets (see Section Inputs and Requirements), she has to click on a single button to activate the scheduling method which generates individual schedules for teachers and students (Table 6) and constructs the whole showcase program.

Schedule for Teacher:	T7		
Heat	**Time**	**Dance**	**Student**
1	3:00:00 PM	American Rumba	S4
4	3:05:00 PM	Argentine Tango	S4
89	4:51:15 PM	American Tango	S2
112	5:20:00 PM	American Tango	S2
198	7:07:30 PM	American Waltz	S15

Schedule for Student:	S1		
Heat	**Time**	**Dance**	**Teacher**
3	3:03:45 PM	American Foxtrot	T9
19	3:23:45 PM	Argentine Tango	T2
68	4:25:00 PM	American Rumba	T2
139	5:33:45 PM	American Rumba	T2
209	7:21:15 PM	American Rumba	T9

Table 6: Schedule for students and teachers. The table on the left side provides the partial schedule for a teacher as it is generated by the Excel decision-making tool. The table on the right side gives the same information for a student.

The implementation of the scheduling method is done in such a way that the Excel scheduling tool is usable by anybody who is comfortable with Excel but who is not necessary familiar with optimization solvers, Visual basic or Macros. The scheduling tool is also very useful in the case of late (up to a reasonable cut-off date) registrations or withdrawals: if necessary, a new schedule taking into account the new conditions can be constructed extremely easily and fast. We use the procedure below to make the changes in the schedule due to the late registrations easier. The procedure first checks whether the optimal total number of Heats remains unchanged or must be increased as a consequence of the late entries. If it must be increased, additional Heats are added at the end of the showcase. This method has the advantage that it does not affect the schedule of any of the other participants and it covers a wide range of possibilities. More concretely, we were able to implement the procedure for the construction of the Summer 2007 showcase without it resulting in the violation of any requirements. In that case, the management of the studio was satisfied with not having the entries of the late registrant spread throughout the four quarters. In the second step of the procedure, we identify a subset of Heats that can be modified to incorporate the entries of the late registrants. The subset is composed of the Heats that (i) feature the types of dances concerned by the late registrations and (ii) are scheduled at times when the late registrants are available. A heuristic is implemented in which the Heats in this subset are successively considered for the incorporation of the late entries and the schedule for all the other Heats remains unchanged. We prioritize each Heat: at each iteration the heuristic considers the not yet considered Heat with the largest priority. The assignment of the priorities obeys the following basic rules: (i) one-couple Heats satisfying the above requirements receive the highest priority; (ii) additional Heats (if any) receive the lowest priority.

As a result of the implementation of the scheduling tool, instead of spending weeks constructing and modifying the schedule by hand, the studio manager can now input the data in an Excel file and run the algorithm with one click of a button which generates a complete schedule in a matter of less than 5 minutes. In case of changes in student preferences and management requirements, the algorithm can be simply rerun to generate a modified schedule corresponding to the new conditions and data. Additionally, an exact time slot is generated for each Heat which helps estimate the total duration of the showcase, the accommodation of students' time requirements, and the insertion of breaks and/or the mini-program of Solo dances. All these improvements have allowed the management to significantly cut the time spent on schedule construction; this time is now devoted to other crucial activities such as customer relations, sales, teaching and teacher training. Moreover, since showcase participations represent significant financial input from students and from the studio, the better arrangement of the schedule would make the showcases more efficient, more compact, and more interesting for the students, thus encouraging their future participation. The scheduling tool turned out to be very successful and was used to generate the program for the Summer 2007 showcase of Fred Astaire East Side Dance Studio, and is currently being used in the preparation of the Winter 2007 showcase. We believe that the management of the studio could be very much interested in integrating additional constraints in the future, allowing for a greater satisfaction of the requirements of their students.

Appendix

HEURISTIC

We provide here a more detailed description of the first step of the algorithm in which we assign a type of dance to each Heat. The total number of Heats is set equal to its minimum whose value r is obtained either through the application of the algorithm or through the solution of MODEL I. The number of Heats for each type d of dance is set equal to $\hat{N}_d$. We proceed as follows:

- we order the types of dance by decreasing order of their corresponding minimal number $\hat{N}_d$ of Heats: $d_1, d_2, ..., d_{|D|} : \hat{N}(d_1) \geq \hat{N}(d_2) \geq ... \geq \hat{N}\left(d_{|D|}\right)$. The notation $|D|$ denotes the cardinality of the set D of type of dances.

- we assign a type of dance to a Heat as follows: $\begin{cases} Heat\ 1 = d_1 \\ Heat\ 2 = d_2 \\ \quad ... \\ Heat\ |\mathrm{D}| = d_{|D|} \end{cases}$

  We repeat this sequence $\hat{N}\left(d_{|D|}\right)$ times until the total number of Heats $\hat{N}\left(d_{|D|}\right)$ of dance type $d_{|D|}$ (the dance type with the smallest number of Heats in the schedule) is assigned:

$$\begin{cases} Heat\ 1 = Heat\ 1 + |\mathrm{D}| = ... = Heat\ 1 + |\mathrm{D}| \times \hat{N}\left(d_{|D|}\right) = d_1 \\ Heat\ 2 = Heat\ 2 + |\mathrm{D}| = ... = Heat\ 2 + |\mathrm{D}| \times \hat{N}\left(d_{|D|}\right) = d_2 \\ \quad ... \\ Heat\ |\mathrm{D}| = Heat\ 2|\mathrm{D}| = ... = Heat\ |\mathrm{D}| \times \left(\hat{N}\left(d_{|D|}\right) + 1\right) = d_{|D|} \end{cases}.$$

  The next assignment sequence will be: $\begin{cases} Heat\ 1 + |\mathrm{D}| \times \left(\hat{N}\left(d_{|D|}\right) + 1\right) = d_1 \\ Heat\ 2 + |\mathrm{D}| \times \left(\hat{N}\left(d_{|D|}\right) + 1\right) = d_2 \\ \quad ... \\ Heat\ |\mathrm{D}| - 1 + |\mathrm{D}| \times \left(\hat{N}\left(d_{|D|}\right) + 1\right) = d_{|D|-1} \end{cases}$

  We repeat this assignment sequence $\hat{N}\left(d_{|D|}\right) - \hat{N}\left(d_{|D|-1}\right)$ times until all Heats of dance type $d_{|D|-1}$ are assigned. We continue this until we reach d_1 (the dance type with the largest number of Heats in the schedule). Then, three distinct cases must be considered:

- if $\widehat{N}(d_1)=\widehat{N}(d_2)$, there is no additional Heat needed;
- if $\widehat{N}(d_1)=\widehat{N}(d_2)+1$, we add one additional Heat of type d_1 (last Heat in the schedule) and the assignment process terminates;
- if $\widehat{N}(d_1)-\left(\widehat{N}(d_2)+1\right)=c>0$, we add one Heat of type d_1 at the end of the schedule and *c-1* additional Heats of type d_1 are added earlier in the schedule with the only requirement that their inclusion does not result in two consecutive Heats of type d_1.

NOTATIONS

We introduce below the notations that are used to formulate the two integer programming problems.

Index Sets

I: set of students; *J*: set of teachers; *D:* set of types of dances; T = set of Heats: $T=\bigcup_{k=1}^{4} T_k$ with T_k representing the set of Heats in quarter *k* (*k*=1,2,3,4); *R = (i,j,d,w)*: set of entries which is a set of 4-tuples where *i*, *j* and *d* respectively refer to the student, teacher, and type of dance and *w* is an index differentiating the duplicate entries (indeed, a student can register to dance several times the same type of dance with the same teacher).

Decision Variables

x[i,j,d,w,t]: binary variable taking value 1 if the registered dance *(i,j,d,w)* is scheduled to take place during Heat *t*, and 0 otherwise.

s[d,t]: binary variable taking value 1 if the Heat *t* is for dances of type *d*, and 0 otherwise.

$\gamma[t]$**:** binary variable taking value 1 if Heat *t* has more than one student-teacher pair scheduled to dance, and 0 otherwise.

$\sigma[i,k]$: binary variable taking value 1 if student *i* has a dance scheduled during quarter T_k, and 0 otherwise.

Parameters

$a[i,t]$*:* indicates whether student *i* is available ($a[i,t]=1$) or not ($a[i,t]=0$) in Heat *t*

$b[j,t]$: indicates whether teacher j is available ($b[j,t]=1$) or not ($b[j,t]=0$) in Heat t

M: maximum number of student-teacher pair dancing per Heat

$W_{i,j,d}$: number of times that student i wants to dance the type d of dance with teacher j

r: minimum total number of Heats

$E(i)$: number of entries for student i

$Q(i)$: number of quarters during which student i is available

MODEL I

Objective Function

MODEL I minimizes the duration of the dance program and therefore minimizes the number of Heats in the showcase: minimize $\sum_{d\in D}\sum_{t\in T} s[d,t]$

Constraints

- Assignment of at most one type of dance per Heat: $\sum_{d\in D} s[d,t] \le 1, t \in T$

- Time availability of students: $\sum_{j\in J}\sum_{d\in D}\sum_{w=1}^{W_{i,j,d}} x[i,j,d,w,t] \le a[i,t], i \in I, t \in T$

Note that the above also enforces that each student is assigned at most once to each Heat (since $a[i,t] \le 1$).

- Time availability of teachers: $\sum_{i\in I}\sum_{d\in D}\sum_{w=1}^{W_{i,j,d}} x[i,j,d,w,t] \le b[j,t], j \in J, t \in T$

Note that the above also enforces that a teacher is assigned at most once ($b[j,t] \le 1$) to each Heat.

- Not having two consecutive Heats with the same type of dance:

$$s[d,t] + s[d,t+1] \le 1, d \in D, t = 1,\ldots,|T|-1$$

- Not assigning a registered dance to a Heat with a non-corresponding type of dance:

$$x[i,j,d,w,t] \le s[d,t], (i,j,d,w) \in R, t \in T$$

- Assignment of each registered dance to a Heat: $\sum_{t\in T} x[i,j,d,w,t] = 1, (i,j,d,w) \in R$

- Not exceeding the maximal number of couples/entries per Heat: $\sum_{(i,j,d,w)\in R} x[i,j,d,w,t] \le M, t \in T$

- Avoiding interruptions (*i.e.*, Heat without any scheduled entry followed by other Heats with scheduled entries) in the program: $\sum_{d\in D}\sum_{t\in T} s[d,t] \geq \sum_{d\in D}\sum_{t\in T} s[d,t+1], t=1,\ldots,|T|-1$

The above constraint ensures that if an entry is scheduled during Heat *t+1* ($\sum_{d\in D}\sum_{t\in T} s[d,t+1]=1$), then all preceding Heats *t' =1,…, t* are not "empty" and have entries scheduled for them too.

- Definition of binary decision variables: $x[i,j,d,w,t], s[d,t] \in \{0,1\}, (i,j,d,w) \in R, d \in D, t \in T$

MODEL II

MODEL II includes two other desirable properties for the schedule (from the point of view of the studio), namely the minimization of the number of Heats with only one student-teacher pair on the dance floor and the involvement of the students during the entire showcase by scheduling each of them to dance throughout the whole event, provided that they are available. The problem remains a binary linear optimization problem. We note that the requirement to keep the students involved across the whole showcase could lead to an increase in the number of one-couple Heats and in the duration of the showcase.

Objective Function

MODEL II maximizes the number of Heats in which there is more than one entry scheduled:

$$\text{maximize} \sum_{t\in T} \gamma[t]$$

Constraints

- Guarantee that the schedule contains a minimum total number *r* (whose value is known through the solution of MODEL I) of Heats: $\sum_{d\in D}\sum_{t\in T} s[d,t] \leq r$

- Constraint defining the value of the decision variable $\gamma[t]$ that indicates whether the number of entries scheduled in any Heat *t* is larger than or equal to 2: $\sum_{(i,j,d,w)\in R} x[i,j,d,w,t] \geq 1+\gamma[t], t \in T$

The constraint ensures that the binary variable $\gamma[t]$ can only take value 1 if the number of entries scheduled in Heat *t* is at least equal to 2. Otherwise, $\gamma[t]$ takes value 0.

- Involvement of students through the whole showcase: each student, if available (and registered for at least 4 entries), has at least one entry in each quarter:

$$\sum_{t \in T_k} x[i, j, d, w, t] \geq \sigma[i,k], i \in I, k = 1,2,3,4$$

$$\sum_{k=1}^{4} \sigma[i,k] \geq \min(Q_i, E_i), i \in I$$

The notation $\min(a,b)$ refers to the smallest value among a and b. The first constraint above forces the binary variable $\sigma[i,k]$ to be equal to 0 if no entry is scheduled for student i during the whole quarter T_k. The second constraint guarantees that each student dances in a number of quarters at least equal to the minimal value of the number of entries of the student and the number of quarters during which the considered student is available;

- Definition of additional decision variables: $\gamma[t], \sigma[i,k] \in \{0,1\}, t \in T, i \in I, k = 1,2,3,4$

- All the constraints that apply to MODEL I if the minimal number of Heats is computed and approximated though the algorithm proposed in the Construction of Showcase Schedule Section. In that case, the above decision variables and constraints must be kept. However, if the true minimum number of Heats is computed (*i.e.*, if we solve MODEL I to optimality), then the following modifications regarding the set of constraints are allowed:

    - the constraints $\sum_{d \in D} \sum_{t \in T} s[d,t] \geq \sum_{d \in D} \sum_{t \in T} s[d,t+1], t = 1,...,|T|-1$ that prevent an interruption (*i.e.*, Heat without an entry) during the program become redundant and can be removed. Indeed, when the exact minimal number of Heats is known (through the optimal solution of MODEL I) and enforced, MODEL II constructs a schedule with minimal number of Heats, thus preventing us from having a Heat without a student-teacher pair scheduled for it;
    - the constraints $\sum_{d \in D} s[d,t] \leq 1, t \in T$ that ensure that at most one type of dance is assigned to each Heat can now be rewritten as $\sum_{d \in D} s[d,t] = 1, t = 1,..., r$ for the same reason that the above constraints can be removed.

SUMMER 2007 SHOWCASE

Characteristics of Summer 2007 Showcase			
Number of Entries: 583 -- Number of Students: 28 -- Number of Teachers: 8 -- Number of Dance Types: 19			
Comparison of Heuristic and Integer Programming Approaches			
	Heuristic	MODEL I	MODEL II
Number of Heats	240	240	240
Number of One-Couple Heats	40	48	39
Computational Time (in CPU seconds)	298 sec	9195 sec	9444 sec
Number of Decision Variables		5143	5495
Number of Constraints		154163	154305

Table 7: Summer 2007 Showcase's Characteristics and Comparison of Solution Approaches. The second row reports the dimensions of the Summer 2007 schedule. Row 5 (respectively 6 and 7) report the number of Heats, one-couple Heats and the computational time to construct the schedule with each of the three considered solution approaches (columns 2, 3 and 4). Rows 8 and 9 indicate the number of constraints and (binary) decision variables in each of the two integer models.

References

Dinitz, J.H., Froncek, D., Lamken, E.R., Wallis W.D. 2006. Scheduling a Tournament. In: Colbourn, C.J., Dinitz, J.H. (eds.), *Handbook of Combinatorial Designs*, CRC Press.

Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D. 2004. Staff Scheduling and Rostering: A Review of Applications. *European Journal of Operational Research* **153** (1), 3-27.

Gilbert, K.C., Hofstra, R.B. 1987. Algorithms for a Class of Three-Dimensional Assignment Problems Arising in Scheduling Applications. *IIE Transactions* **19** (1), 29-33.

Trick, M. 2008. *Michael Trick's Guide to Sports Scheduling*. Retrieved January 27, 2008 from *http://mat.gsia.cmu.edu/sports/* .