Proceedings from 32nd International Symposium on Military Operational Research
21-24 July 2015, Royal Holloway, University of London, Egham, Surrey, United Kingdom

Course loading optimization

Eisler, C. [1], Dobias, P. and Lu, O.

Defence Research and Development Canada – Centre for Operational Research and Analysis
101 Colonel By Dr., Ottawa, ON, K1A0K2, Canada

Abstract

As of 2014, the Naval Personnel Training Group (NPTG) became responsible for all Royal Canadian Navy (RCN) schools. The initial review conducted by the NPTG staff looked at the staff-to-student ratios, which revealed that there were potential inefficiencies that may have increased the cost of school operations. The authors were asked by the NPTG to conduct a detailed review of the course loading, defined as the number of students that are enrolled for a given course, and identify optimal loading to ensure that a) the RCN personnel gets the required training, b) the course loading is within the estimated minimum and maximum number of required students for each course, and c) the overhead in terms of staff and resource requirements is minimized. The analysis was conducted in two phases. In Phase 1, the main inefficiencies were identified; these included duplicate and excess courses, and courses with insufficient enrollment. The results proposed restructuring the offered courses in order to meet the demand while eliminating excess. The minimum and maximum enrollment for each course was combined with the historical data to obtain these results. In Phase 2, the course loading was further optimized with respect to resource utilization while considering constraints in terms of student demand and resource supply.

Introduction

An analysis of the Royal Canadian Navy (RCN) school course attendance data from 2010-2014 revealed that, for many courses, the supply (in terms of offered sessions) surpasses the expected demand based on the minimum required student load. Consequently, the Naval Personnel Training Group (NPTG) approached the Maritime Command Pacific (MARPAC) Operational Research Team (ORT) to assist in analyzing the supply and demand relationship

[1] cheryl.eisler@drdc-rddc.gc.ca

and to identify optimal number of courses that would meet student expected demand while minimizing course cancellations. The study was completed in two phases. In Phase 1, the course minimum and maximum load were used to estimate bounds on the number of offered sessions required to meet the historic student demand [1]. This part of the analysis was straightforward; the minimum required course load was employed to provide the estimate for the maximum number of sessions required to meet the estimated demand. For example, if a course is required to have at a minimum 5 students, and there are on average 18 students a year taking the course, the maximum number of sessions for this course would be 3. The maximum number of students allowed per course could then be used to determine what the minimum number of sessions required to meet the expected demand. However, this approach ignores possible additional constraints, such as the availability of students, availability of resources, and required sequencing of courses (e.g., if the course *A* is a prerequisite of the course *B*, students cannot take *B* before they take *A*). Consequently, a more robust approach was needed.

Phase 2 of the study is described this paper. It looks at the use of genetic algorithms and Monte Carlo simulations to refine the estimate of the course offering requirements. The paper is organized as follows. The scheduling optimization model is presented first, describing the general approach to the problem. The individual components of the model are then explained in more detail. Sample results from the model are utilized to illustrate the functionality of tool that was created and highlight areas for future development. The analysis presented in this paper served only as a test bed for the optimization model and consequently relied solely on the historical data. However, in order to implement the model as a client support tool the expected student demand would need to be validated by the NPTG staff. Additional factors going well beyond the information obtainable from the historical data such as changes to study program, policies, requirements, as well as manpower and resource availability will need to be considered. Therefore the anticipated course load will need to be based on a blend of historical and forecasted data.

The scheduling optimization model

The optimization model was built in Microsoft Excel utilizing the Solver add-in, in order to facilitate its employment on the defence network. The problem was separated into two steps:

- Estimating the required course demand; and,
- Course scheduling and resource assignment.

Under the assumption that the historical demand for students is a valid predictor of future demand[2], the required course estimate was generated based on student demand obtained from

[2] This might not be the case for a real world application, as is discussed in Introduction.

a course database maintained by the NPTG, combined with Monte-Carlo Simulation (MCS) optimization with respect to a single objective. The course scheduling problem is a modified version of the classical Job Shop Scheduling Problem (JSSP), defined as the task of scheduling *N* tasks of varying durations to *M* identical machines to be completed presented by Graham [2]. However, the RCN school student demand problem differs slightly from the original JSSP, since classes require multiple resources that are generally of different nature (e.g., staff, classrooms, ships, or training simulators). Any resource that is assigned to one course cannot be simultaneously assigned to any other courses; therefore, the model must be able to accommodate insufficient resources to meet the course demand.

Figure 1 is a diagrammatic representation that summarizes this process in a flow chart. Course information, such as the maximum and minimum class sizes, is combined with the historical student course demand in a MCS. This process generates the optimal number of sessions for each course. Combined with the appropriate resource constraints, this is then passed into a resource assignment algorithm. After all of the assignments have been made, a fitness score is computed, and a new schedule chosen using an evolutionary algorithm. The process of assigning resources repeats until a user-defined end condition is achieved (such as goodness of fit, maximum run time, etc.).

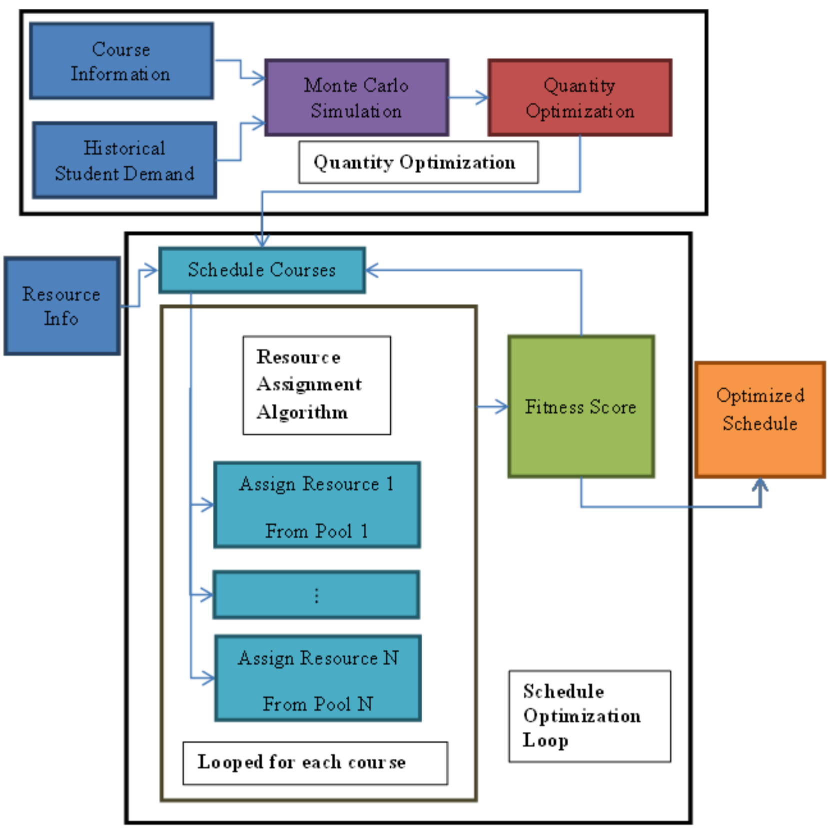

Figure 1: *Representation of the naval schools course scheduling optimization algorithm.*

It is important to note that this initial model does not take into account location (i.e. there are three main locations for the Canadian naval schools, and staff and students can move among them), cost, or human factors (such as the impact of course loading on learning). In addition, the examination of the ideal minimum and maximum number of students for individual courses is not within the scope of this study.

Estimating course demand

One of the primary considerations for the course scheduling is the year-to-year variation in student enrollment. To determine most likely student demand, the 2010-2014 historical course attendance data were used. A probability distribution was generated on a course-by-course basis for the demand and used to calculate the optimum number of sessions for each course with respect to an objective function. Here, the objective function was defined to be the fractional capacity of one course, f, where n_s is the number of students that wish to take the course and n_{max} is the maximum number of students allowed in the course, such that:

$$f = n_s/n_{max}$$

The mean value of f, normalized over all courses, would then be maximized. A penalty function quantifying undesirable outcomes (e.g., the number of courses cancelled, or added) is then considered as a constraint. If h is the number of extra courses that must be added, k is the number of cancelled courses, and a is some pre-defined constant, the penalty (constraint) function could be defined as:

$$c = h + k < a$$

In this case, the constraint function c is used to describe oversupply and undersupply as equally undesirable. The process is summarized by the “Quantity Optimization” module in the algorithm diagram in Figure 1.

As is mentioned above, five years of annualized data were available for the course loading. In order to calculate the future demand, it was assumed that only the actual (non-zero) student loading was valid. In addition, some courses were not offered every year. As a consequence, for each course there would be between one and five data points. This limited statistical validity of the load values and constrained the load fitting options. In order to deal with this limitation, the following methods were used to generate probability distributions. For courses with:

- One (1) data point: No distribution was used, the single data point was directly included in the scheduling module;
- Two (2) data points: A uniform distribution between the two sample values was used;
- Three (3) or four (4) data points: A triangular distribution was used; and,
- Five (5) data points: A distribution fit to the data using the @*Risk6* distribution fitting tool was used, as this tool requires at least five data points.

These expected student load estimates were then used to calculate the upper and lower bound for the number of sessions required for each course as follows. The maximum theoretical number of sessions that can be offered without cancellations is simply the number of students divided by the minimum class size. Similarly, the minimum theoretical number of courses required to enroll all students is the number of students divided by the maximum class size; this number is increased by one (1), if the remainder meets the minimum load constraint. However, due to the stochastic nature of the expected student load, these numbers have additional uncertainty connected with them. In order to better estimate of the risk of having to cancel or add more sessions, the *@RISK Optimizer* engine[3] was used to generate the required number of course sessions, with the expected student load probability distributions serving as an input to the MCS.

In order to test the approach, the MCS optimization was run for the sample size *i* = 5000 occurrences of a course *X,* for which the annualized load data were available for all five years. The results for the course fractional capacity (as defined previously) are shown in Figure 2; in this case, the extreme value probability distribution function provided the best fit,[4] with the normal distribution being fairly good fit as well. From both distributions the mean fractional capacity, <*f*> ≈ 0.74 +/- 0.06 (with ~90% confidence); or alternatively, there is ~5% probability that the fractional capacity would be below 0.67 (i.e., less than 2/3 of available seats for this course would be filled), and about 5-6% probability that the fractional capacity would be greater than 0.81 (i.e., 4/5 of the seats would be taken).

Thus, as the number of sessions of a course tends towards the minimum theoretical number required to enroll all students, the mean fractional capacity increases. However, as long as the constraints are satisfied, such that the number of sessions of one course is greater than the minimum and less than the maximum specified (within a given iteration), no new sessions need to be added or existing sessions need to be cancelled. This is because the course loading can be rebalanced to ensure that the minimum number of students enroll in each course; the fractional capacity will decrease.

---

[3] The @Risk Optimizers combines Monte-Carlo simulations with a genetic algorithm to arrive at an optimal solution in the presence of variables whose exact dynamics are unknown. (*RiskOptimizer, User Guide, by Pallisade Corp.)*

[4] According to chi-square goodness of fit test utilized within @*Risk*.

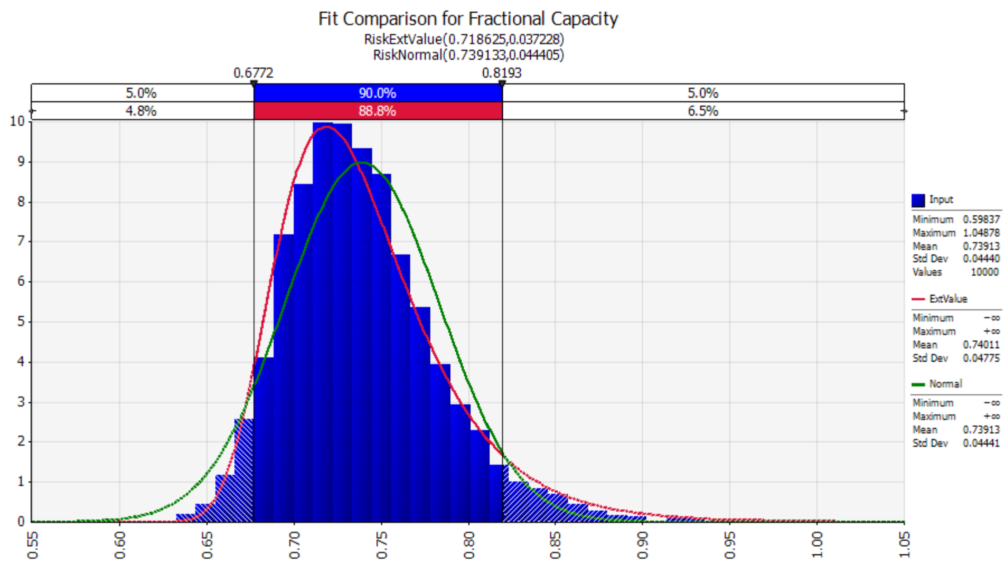

Figure 2: *Example of @RISK probability distribution fit for the fractional capacity of course X.*

Scheduling module

Given a number of courses (with defined durations) and a pool of resources, the schedule optimization problem can be summarized as finding the best choice of start dates, while considering resource assignment limitations, with respect to some objective fitness function. Wesolkowski, et al. [3] demonstrated that one way to estimate the resource allocation in a similar problem is to compute a mean average of the demand duration $d_c(r)$ for a resource r, used by course c. In this approach, the start dates are randomly generated and the total duration is calculated for each resource; these durations are then averaged for further use.

The fitness function would incorporate cost and schedule conflicts (if any), e.g., in the form of a weighted sum objective [4]. The weighted sum approach was selected because the Microsoft Excel Solver add-in, developed by Frontline Systems and used for this project, is incapable of defining multiple objective problems. The weighted sums approach can be used to overcome this limitation. The start dates can then be specified as input variables for which the objectives can be optimized. A resource selection framework within the simulation chooses the resource candidates to be assigned to certain dates; once all assignments are made the fitness objective can be computed (Figure 1; *Resource Assignment Algorithm*). After the fitness is computed, an optimization loop continues using the Frontline Systems Evolutionary Algorithm [5] until predefined termination conditions are reached (such as the maximum number of sub-problems or time spent on computation with no improvement to the fitness function). For any genetic algorithm, such termination parameters must be specified because, unlike in gradient-based or combinatorial optimization methods, there is no notion of derivatives or convergence.

To assess the goodness of fit, a scoring function, based on a modified variable transformation of the Weibull Probability Density Function (WPDF) was used. The WPDF was selected solely because of its shape; there are no fundamental reasons behind the selection. The distribution can be formulated as:

$$w_t(x) = \alpha\beta^{-\alpha}(-(x_t - h))^{\alpha-1} e^{-\left(\frac{-(x_t-h)}{\beta}\right)^{\alpha}}$$

where *x* is the scoring parameter and α, β, and *h* are fitting parameters; for this analysis, the fraction of schedule slots that are currently assigned to a course was the only parameter employed.

In order to help the Solver select feasible solutions, the concept of localized conflicts (LCs) is introduced. It was assumed that a given resource was not able to complete multiple tasks at once (i.e., a staff member cannot teach more than one course at a time). The total number of scheduling conflicts can be written as:

$$C(\mathbf{A}) = \sum_{i=1}^{m}\sum_{j=1}^{n} c(\mathbf{A}(i,j))$$

where **A** is the schedule matrix for all resources across time (with the element value indicating the number of scheduled courses for any given day), and *c* is the number of conflicts, such that:

$$c(x) = \begin{cases} x - 1; x \geq 1 \\ 0;\ x = 0 \end{cases}$$

The problem with this formula is that, if used directly for scoring, it does not score a completely full schedule (without conflicts) any better than a completely empty schedule. Therefore, the number of conflicts was used only to test feasibility of a solution and not the actual value of the solution. Instead, the schedule fractional capacity, f_{schedule}, was used as the fitness function (i.e., the measure of how full the schedule is).

$$f_{\text{schedule}} = t/T$$

where t is the total duration of all courses assigned to a resource and T is the total number of available schedule slots. This fitness function is averaged over all resources to provide a single summary number. It must be noted that it is possible for the fractional capacity to yield values greater than one; this would imply that additional resources would be required to handle the demand.

In terms of the penalty/fitness functions, the current model treats all courses in the same way. It does not distinguish between different costs of different courses. In other words, a course requiring expensive infrastructure is treated in the same way as a simple classroom course. These different costs could be reflected in a more complex penalty and fitness functions. However, because the present model is intended only as a test, the more complex approach was not pursued. If a high fidelity scheduling model is required, or if actual monetary cost analysis is required, the penalty function will have to be readjusted.

A test simulation was then run, considering 40 courses over 50 week time span, with 113 resources separated into 8 pools. The resource data were obtained from the school databases, where only critical resources were considered. After running the optimization using the standard Solver, an optimal solution with no schedule conflicts (i.e., a feasible solution) was found, with the fractional capacity $f_{schedule}$ = 0.37. Table 1 shows a sample of the results for several resources, and Table 2 selects a single resource and indicates specific scheduling assignments. Table 3 shows a snapshot of a part of the overall schedule.

Table 1: *Resource summary examples.*

Resource Type Number (Quantity Number*), Pool	Fractional Capacity	Conflicts
R39,6	0.79	0
R11(1),3	0.77	0
R27,4	0.75	0
R44(1),6	0.68	0
R46,6	0.67	0
R59(1),7	0.65	0
R29(1),5	0.65	0
R48(1),6	0.59	0

* Optional (Indicated only if there is more than one of a given type)

Table 2: *Example of one resource assignment.*

R39,6

Class Code	Start Date	Duration
ABCD	5/22/2015	102
ABCE	1/23/2015	95
ABCF	9/19/2015	80

Table 3: *Snapshot of the resulting example schedule for a given resource.*

R39,6

Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1/1/2015	0	0	0	0	0	0	0
1/8/2015	0	0	0	0	0	0	0
1/15/2015	0	0	0	0	0	0	0
1/22/2015	1	1	1	1	1	0	1
1/29/2015	1	1	1	1	1	1	1
2/5/2015	1	1	1	1	1	1	1
2/12/2015	1	1	1	1	1	1	1
2/19/2015	1	1	1	1	1	1	1
2/26/2015	1	1	1	1	1	1	1
3/5/2015	1	1	1	1	1	1	1
3/12/2015	1	1	1	1	1	1	1
3/19/2015	1	1	1	1	1	1	1
3/26/2015	1	1	1	1	1	1	1

Because the presented results were intended only as a test (and demonstration of feasibility) of the methodology, and were based only on a subset of the actual course list with some of the data artificially generated, there was no comparison of the results with the actual schedule conducted. In order for the comparison to be meaningful, more complete data, and a complete course list would have to be used. However, the results were compared to the first-look analysis [1]. The latter did not consider constraints of limited resources and looked solely at the supply and demand ratios. Therefore, the present analysis was an important validation of the feasibility of the savings proposed in the results of the first-look analysis. In fact, the present model suggested that further minor savings were feasible.

Similarly, a rigorous sensitivity analysis was not done on the present model. However, a somewhat random check of the results under varying student demand was conducted, and it suggested that the results would be somewhat stable. However, a proper sensitivity analysis will be required before operationalizing the results in order to ensure that the outcome is not over-optimized and consequently likely to be unstable and with no resilience in the face of real-world uncertainties.

Summary and Conclusions

An *MS Excel* model was developed that identifies the optimal number of sessions meeting expected student demand while minimizing surplus and subsequent session cancellations, all while considering resource availability. The model was designed to provide relatively low-fidelity estimates concentrated on the availability of critical resources only. Since the key parameters were not available for all the courses, only preliminary tests were conducted, partially using artificially generated data, in order to test the feasibility of the approach for the problem with a realistic size.

The model performed well for the test case discussed here; it was able to find a feasible solution for a limited scope problem (in terms of the number of courses and resources). This was in part enabled by using a resource assignment algorithm, which reduces the problem size by eliminating the combinatoric nature of the resource selection. The overall run duration was only approximately five hours on a regular personal computer. The main limitation of the existing model is the maximum of 200 courses that can possibly be scheduled by the optimization engine. This limitation is due to the 200 variable limit of the Microsoft Excel Solver add-in [6], and could be remedied by using a different solver engine.

When compared to a first-look analysis of the course supply-demand relationship [1], the present model yielded very similar results. It suggested that further minor improvement in efficiency relative to the initial analysis were possible. What is more important, the first-look analysis did not consider resource constraints; therefore the presented results provided an important validation of the feasibility of the optimized solutions with regards to the resources.

The use of historical data to model future demand may lead to some inaccuracies. For example, for a brand new (or recently changed) course there may no historical data for previous years, and even if there are, they are unlikely to be a true reflection of future course needs. Alternatively, some past courses may have been discontinued because they would not reflect future requirements. Therefore, for actual operational use of the tool, the student demand would need to be a combination of historical and forecasted data, and would have to be validated by the NPTG staff.

Additional significant limitation of the current model is that it does not distinguish between different costs of different courses. In other words, a course requiring expensive infrastructure is treated in the same way as a simple classroom course. These different costs might be reflected in the penalty function used for the optimization algorithms; however, to do so was beyond the scope of the test case presented here.

Furthermore, before the model can be operationalized, additional tests, including detailed and rigorous sensitivity analysis, will be required to ensure robustness of the results under real-world conditions.

References

1. Eisler, C., Lu, O., and Dobias, P. (2015), Private communication to NPTG.
2. Graham, R. (1966). Bounds for Certain Multiprocessing Anomalies. The Bell System Technical Journal, 1563-1581.
3. Wesolkowski, S., Wojtaszek, D., Eisler, C., and Stuive L. (2011), Maritime Fleet Mix Optimization: Using the SaFE Model to Search for Fleets that are Non-Dominated in Size, Vignette Duration, and Risk, (DRDC CORA TM 2011-199), Defence R&D Canada – Centre for Operational Research and Analysis.
4. Kim, I. Y., and de Weck, O. L. (2005, February). Adaptive weighted sum method for multi-objective optimization: a new method for Pareto front generation. Structural and Multidisciplinary Optimization, 31(2), 105-116.
5. Frontline Systems Inc. (n.d.). Genetic Algorithms and Evolutionary Algorithms. Retrieved April 13, 2015, from Frontline Solvers: http://www.solver.com/genetic-evolutionary-introduction.
6. Frontline Systems Inc. (n.d.). Standard Excel Solver - Dealing with Problem Size Limits. Retrieved April 10, 2015, from Frontline Solvers: http://www.solver.com/standard-excel-solver-dealing-problem-size-limits#Limits on Decision Variables.