Resource Scheduling for the United States Army's Basic Combat Training Program

Michael L. McGinnis
Operations Research Center, U.S. Military Academy, West Point, NY 10996

Emmanuel Fernández-Gaucherand
Systems and Industrial Engineering, University of Arizona, Tucson, AZ 85721 email: emmanuel@sie.arizona.edu

1. Introduction

Each year, the United States Army recruits and trains thousands of soldiers to fill vacancies in Army organizations. Installations responsible for training new recruits are scattered across the United States. Initial entry training for new recruits is conducted in two phases: Basic Combat Training (BCT) followed by Advanced Individual Training (AIT).

Proper management of the Army's initial entry training program is a very complex, practical military logistics problem that demands timely scheduling of a broad range of reusable training resources, such as, training companies. Currently, manual heuristic methods are used to schedule training companies throughout the planning horizon to support initial entry training, where training company scheduling also involves deciding how many recruits to assign to training companies each week. There are several severe shortcomings with these methods. For example, determining the number of recruits assigned per training company and the number of weeks a training company remains busy training recruits is a manual trial-and-error process. Second, it is possible for different analysts to generate different solutions for the same recruitment scenario. Third, no methods exist for conducting comparative analyses to appraise the quality of competing feasible training schedules. Finally, the temporal interdependence of decisions makes decision variables in the future periods depend on current decision variables. This complicates resource scheduling and makes the manual generation of week-by-week training schedules a tedious, time-consuming task.

In this paper we present the following: (1) a mathematical dynamic model of BCT; (2) a decision model for optimally scheduling training resources based on dynamic programming; and (3) an improved automated heuristic procedure for scheduling training resources that incorporates a "training quality" performance measure into the formulation of the objective function, making it possible to compare competing feasible training schedules obtained by various methods. Computational experiments reveal that the heuristic procedures developed are indeed computationally efficient and provide "good" solutions in terms of training "quality," resource utilization, and training cost. The reader is referred to McGinnis [8] for further details and results.

Previous work tangentially related to our approaches include: (1) Yang and Ignizio [11] who developed a heuristic procedure for a (somewhat) similar military training scheduling problem, (2) Wagner and Whitin's [10] formulation of the economic lot-sizing problem (ELSP) for manufacturing and inventory processes and extensions and solution methods to the ELSP (see, for example, [1],[3],[4],[5],[6]), and (3) Rao [9] who developed a manpower planning model that uses manpower requirements for future periods to minimize system costs for a number of fixed and variable recruitment costs.

2. Model Formulation

Before presenting the mathematical model of the basic training problem, practical aspects of Basic Combat Training essential to the proper development of the dynamic training system model are discussed.

Estimating the Weekly Arrival of New Recruits

For the version of the basic training problem presented here, recruit arrivals are estimated ahead of time for each week t and year j of the planning horizon T_J, given an annual recruiting target for each year j. Therefore, the absence of a random disturbance makes this formulation of the basic training problem completely deterministic.

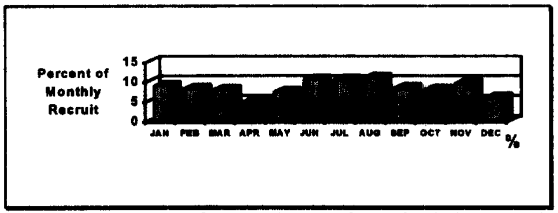

Figure 1. Percent of Annual Recruit Arrivals by Month

Recruiting continues throughout the year and focuses heavily on young people in their final year of high school.

0-7803-2129-4/94 $3.00 © 1994 IEEE

Analysis of the historical data reveals that although annual recruiting targets vary from year-to-year, the distribution of recruit arrivals across the year remains relatively stationary; see Figure 1.

Dynamics of Varying Training Company Strength

Company strength, the number of recruits assigned per training company, is bounded below at 150 and above at 250 recruits. In practice, companies scheduled to start training in the same week are initially assigned equal strengths to simplify logistical problems of training program management. This rule is incorporated into our basic training model. Determining the company strength for a given week t requires the following information:

1. the number of recruits that report for training in week t; and
2. the number of training companies available at the beginning of week t to start training new recruits.

However, the number of training companies available to start training in week t depends upon past company strength decisions. In any week, it is possible for previous weeks' company strength decisions to cause a *training company shortfall*, where the number of training companies is not sufficient to handle the arrival of new recruits (or *training load*).

Instructor-to-Student Ratio

Basic training is, by design, highly stressful for new recruits; an aspect of training that often has a negative effect on recruit learning and retention. Training managers rely on quality instruction and close supervision of recruits to offset some effects of stress. In practice, training managers attempt to keep the *instructor-to-student ratio* around 1-to-16 (or lower if possible) resulting in company strengths of approximately 200 recruits per company. This ratio is used as the primary performance measure in the formulation of both the optimal decision model and the automated heuristic scheduling methods presented here.

Compressing-the-Load

Each week, recruits who report to basic training installations are assigned to training companies. This *fill week* runs from Saturday through midnight Thursday. Basic Combat Training begins on Friday and, in general, lasts eight weeks. Normally, a *maintenance week* is scheduled before the next training cycle begins. This ten-week sequence is called a *normal training cycle*.

In some cases, it may not be possible to eliminate a training company shortfall by adjusting company strengths alone. Another way to correct a training company shortfall is by shortening the training cycles for companies that started either eight or nine weeks earlier. This is done by eliminating either the fill week or the maintenance week, or both. However, this practice, called *compressing-the-load*, can have a negative impact on training company cadre by shortening or eliminating their break between training cycles. Therefore, compressing-the-load is only used when the demand for training companies cannot be met by adjusting company strengths.

Mathematical Notation

j: year of the planning horizon, $j \in \{1,2,\ldots,J\}$;

t: week of a given year j, $t \in \{1,2,\ldots,T_j\}$ where T_j is the number of weeks in year j;

$\delta(t)$: recruit show rate for week t where $0 \leq \delta(t) \leq 1$;

$p(t)$: relative frequency distribution of recruit arrivals over week t of any year;

D_j: number of training companies deactivated in year j;

M_j: number of training companies available at the beginning of year j;

R_j: recruiting objective for year j determined by Department of the Army;

$d_j(t)$: number of training companies to deactivate in week t of year j;

$r_j(t)$: estimated number of recruits that show up for training in week t of year j;

$x_j(t)$: strength of companies starting in week t of year j;

$\bar{X}$: upper bound for training company strength;

$\underline{x}$: lower bound for training company strength;

$y_j(t)$: training cycle length for companies starting in week t of year j;

$\bar{Y}$: upper bound for training cycle length;

$\underline{y}$: lower bound for training cycle length;

$D_j^*(t)$: balance of training companies left to be deactivated as of week t of year j;

$I_j(t)$: number of idle training companies at the beginning of week t of year j;

$\bar{I}$: upper bound for the idle training company constraint.

Modeling Constraints

$\underline{x} \le x_j(t) \le \overline{X}$: company strength constraint; (1)

$\underline{y} \le y_j(t) \le \overline{Y}$: training cycle constraint; (2)

$M_j \ge D_j$: deactivation scenario constraint; (3)

$d_j(t) \ge 0 \ \forall \ (t,j)$: company deactivation constraint; (4)

$0 \le I_j(t) \le \bar{I} \ \forall \ (t,j)$: problem feasibility constraint; (5)

$r_j(t) = \delta(t)\, p(t)\, R_j$,: expected number of (6)
recruits to arrive for training in week t of year j, where the recruiting objective R_j, determined by the Department of the Army (DA), is greater than the number of new soldiers required to meet the needs of the Army in a given year j.

3. Optimal and Heuristic Decision Processes

3.1 Optimal Decisions Using Dynamic Programming

We remove the training company deactivation decision from the DP formulation of the basic training problem by requiring training company deactivation decisions, $d_j(t)$, to be made prior to implementing the DP algorithm.

Stages

In the basic training problem, stages are specified by week t and year j. The planning horizon T_j consists of a finite number of identical, discrete time periods where $t \in \{1,2,\ldots,T_j\}$ and $j \in \{1,2,\ldots,J\}$.

Decisions, Scheduling Policy and Objective Function

In the basic training problem, company strength $x_j(t)$ and training cycle length $y_j(t)$ decisions are made at the beginning of period t for $t = 1, 2,\ldots,T_j - 1$ for all training companies that begin training that period. A sequence of such decisions, denoted by π, is represented by

$$\pi = \begin{Bmatrix} x_1(1),\ x_1(2),\ldots,x_J(T-1); \\ y_1(1),\ y_1(2),\ldots,y_J(T-1) \end{Bmatrix},$$

and the set of all such feasible sequences (i.e., those satisfying (1) - (5)) will be denoted by Π. Here we are interested in obtaining an optimal training schedule that maximizes the "quality" of training. When each training company is of equal size, then maximizing the instructor-to-student ratio (i.e., minimizing company strengths) is equivalent to minimizing idle training companies. If we assume one instructor per training company for simplicity, then for each sequence of decisions $\pi \in \Pi$, a corresponding value J_π, which provides a measure of quality to be maximized, is given by

$$J_\pi = \sum_{t=1}^{T_j-1} \frac{1}{x_j(t)}. \qquad (7)$$

The optimal sequence of decisions π^* is the one that maximizes the following objective function (based on the instructor-to-student ratio) for a fixed initial state

$$J_{\pi^*} = \max_{\pi \in \Pi} J_\pi. \qquad (8)$$

We require that $x_j(t) \in \Omega$ and $y_j(t) \in \Lambda$, where the decision spaces Ω and Λ consist of the bounded sets of integers specified by the company strength constraint, $\underline{x} \le x_j(t) \le \overline{X}$, and the training cycle constraint, $\underline{y} \le y_j(t) \le \overline{Y}$. The subsets of feasible decisions to take at each stage t are denoted by $\mathcal{X}_j\left[t,\ I_j(t)\right] \subset \Omega$ and $\Upsilon_j\left[t,\ I_j(t)\right] \subset \Lambda$, where feasible decision elements belonging to these two subspaces depend on both the stage t and the state $I_j(t)$ of the basic training system.

State Transition Equation

We have the following balance equation for idle training companies:

$$I_j(t+1) = f_j\left[t,\ I_j(t),\ x_j(t)\right] = I_j(t) + \sum_{l \in L} \frac{r_j(t-l)}{x_j(t-l)} - \frac{r_j(t)}{x_j(t)},$$
(9)

where $f_j\left[t,\ I_j(t),\ x_j(t)\right]$ is explicitly defined as an equivalent representation of the right hand side of (9).

$\frac{r_j(t)}{x_j(t)}$ is an estimate of the number of companies to begin training in week *t* of year *j*, where $r_j(t)$ gives the expected number of recruits to arrive for training each week.

$\sum_{l \in L} \frac{r_j(t-l)}{x_j(t-l)}$ represents the number of companies that become available at the beginning of week $t+1$ to start (another) training cycle having just completed one that began either eight, nine, or ten weeks earlier (see *Compressing-the-Load*). The possible values for $l \in L$ are contained within the set $L \in \{\ (10),\ (10,9),\ (10,9,8)\ \}$ (see [8], Section 2.3, for further details).

To maintain the integer value of $I_j(t+1)$ in (9), we round as follows:

$$\text{For } x_j(t) < \overline{X}:\ \ I_j(t+1) = I_j(t) + \left\lfloor \sum_{l \in L} \frac{r_j(t-l)}{x_j(t-l)} \right\rfloor - \left\lfloor \frac{r_j(t)}{x_j(t)} \right\rfloor; \qquad (10)$$

$$\text{For } x_j(t) = \overline{X}:\ \ I_j(t+1) = I_j(t) + \left\lceil \sum_{l \in L} \frac{r_j(t-l)}{x_j(t-l)} \right\rceil - \left\lceil \frac{r_j(t)}{x_j(t)} \right\rceil. \qquad (11)$$

When the current company strength constraint is *tight* (i.e., an equality constraint) at the upper bound, then the fractional part of the training company is rounded up, as denoted by the ceiling operator $\lceil * \rceil$. When training companies are at full strength, those recruits that are represented by the fractional part of a training company can only begin basic combat training if an additional training company is scheduled to start. In all other cases, the fractional part may be dropped, as denoted by the floor operator $\lfloor * \rfloor$, since, in general, sufficient training spaces will be available in training companies not filled to capacity. For simplicity, the floor and ceiling operators of (10) and (11), respectively, will not be repeated for every future reference to training company computations. However, it is to be understood that these rules are in effect throughout the paper unless stated otherwise.

Company strength $x_j(t)$ and cycle length $y_j(t)$ decisions depend upon past information that cannot be summarized in $I_j(t+1)$ alone. Additional information is made available through state augmentation (see [2]). Including additional variables to the problem can significantly increase both the number of computations required to generate an optimal solution, and the amount of computer memory required. Therefore, the state space is augmented by only the <u>minimum</u> number of variables necessary to make a decision in each period. For the case where training cycle length is fixed at $y_j(t) = 10$ creating a nine-period time lag (thus eliminating training cycle length as a decision), the minimally augmented system is given by

$$\begin{bmatrix} I_j(t+1) \\ s_j^1(t+1) \\ s_j^2(t+1) \\ s_j^3(t+1) \\ s_j^4(t+1) \\ s_j^5(t+1) \\ s_j^6(t+1) \\ s_j^7(t+1) \\ s_j^8(t+1) \\ s_j^9(t+1) \end{bmatrix} = \begin{bmatrix} I_j(t) + \frac{r_j(t-9)}{s_j^1(t)} - \frac{r_j(t)}{x_j(t)} \\ s_j^2(t) = x_j(t-8) \\ s_j^3(t) = x_j(t-7) \\ s_j^4(t) = x_j(t-6) \\ s_j^5(t) = x_j(t-5) \\ s_j^6(t) = x_j(t-4) \\ s_j^7(t) = x_j(t-3) \\ s_j^8(t) = x_j(t-2) \\ s_j^9(t) = x_j(t-1) \\ x_j(t) \end{bmatrix}. \qquad (12)$$

One can think of $\{\ s_j^1(t),\ s_j^2(t), \ldots, s_j^9(t)\ \}$ as "registers" for temporarily storing the required information as the system evolves; see [2] and [8]. Hence, $\{\ I_j(t),\ s_j^1(t),\ s_j^2(t), \ldots, s_j^9(t)\ \}$ constitutes the state of the system in our formulation.

A combinatorial explosion of the *state space* also occurs when attempting to obtain an optimal solution to the real-world basic training problem using dynamic programming. The <u>augmented</u> state space for a single period of the basic training problem (for 1988 training data) requires enumeration of the following state variables:

- number of idle training companies $I_j(t+1)$: 130;
- company strength values $x_j(t-1), \ldots, x_j(t-9)$ for the augmented state: $(101)^9$;
- training cycle lengths $y_j(t)$: 3.

This generates an (upper bound) estimate of the size of the state space, for each period *t*, of $3 x 130 x (101)^9$, or approximately $4.27 x 10^{20}$ (427 million trillion) possible states. Although dynamic programming substantially reduces the amount of enumeration required to obtain an optimal solution by (1) avoiding decision sequences that cannot possibly be optimal and (2) solving the problem one stage at a time, the potential size of the augmented state space for the real-world problem, or for a reduced problem (see Table 1 below) remains quite large. Other exact methods (e.g., integer and mixed integer programming, and complete enumeration) suffer from similar problems; see [8]. These motivated the development of efficient heuristics.

3.2 Heuristic Approaches

The heuristic procedure presented here consists of two heuristics applied in three phases. Phase I starts with an initial training requirement for each week t, denoted by $\{ r_1(1), r_1(2), \ldots, r_j(T) \}$, that is estimated from the initial recruiting target R_j for each year j (see (6)). An efficient *single-pass heuristic* (SPH) makes one forward pass through the planning horizon applying a policy iteration algorithm a finite number of times in each period t until an *initial feasible training resource schedule* is obtained (if one exists) for the currently available resources. The training resource scheduling policy in Phase I is the sequence of decisions on company strength $x_j(t)$ and training cycle length $y_j(t)$ for each period t. The policy is specified by

$$\pi^1 = \left\{ \begin{array}{l} x_1^1(1),\ x_1^1(2), \ldots, x_j^1(T-1); \\ y_1^1(1),\ y_1^1(2), \ldots, y_j^1(T-1) \end{array} \right\},$$

where superscript 1 denotes Phase I.

Phase II considers options for changing the level of resources (e.g., deactivating training companies) available to train recruits, and is motivated by recent decisions to downsize the training installation complex. Our model is restricted to one type of training resource (i.e., training companies), and to decisions that reduce the level of available resources. However, the model is easily modified to also consider resource level increases and multiple reusable resources. If no resource changes are needed, then Phase II may be omitted. The resource scheduling policy for Phase II is

$$\pi^2 = \left\{ \begin{array}{l} x_1^2(1),\ x_1^2(2), \ldots, x_j^2(T-1); \\ y_1^2(1),\ y_1^2(2), \ldots, y_j^2(T-1); \\ d_1^2(1),\ d_1^2(2), \ldots, d_j^2(T-1) \end{array} \right\},$$

where $d_t^2(t)$ is the company deactivation decision in period t, and the superscript 2 denotes Phase II.

Experiments have shown that it is possible to improve resource schedules obtained via SPH by making additional passes through the planning horizon using a modified policy improvement step to further decrease training company strengths. This observation led to the development and implementation of a *multi-pass heuristic*. The multi-pass heuristic (MPH) that improves the resource scheduling policies obtained from the single-pass heuristic.

Phase III uses the initial feasible schedule from Phase II (or from Phase I if Phase II is omitted) as its starting point. The initial company strength scheduling policy is iteratively revised, period-by-period, using MPH that works sequentially backward through the planning horizon until no further improvements to the objective function are possible with the MPH. The final resource scheduling policy, obtained at the completion of Phase III, is given by

$$\overline{\pi}^3 = \left\{ \begin{array}{l} \bar{x}_1^3(1),\ \bar{x}_1^3(2), \ldots, \bar{x}_j^3(T-1); \\ y_1^2(1),\ y_1^2(2), \ldots, y_j^2(T-1); \\ d_1^2(1),\ d_1^2(2), \ldots, d_j^2(T-1) \end{array} \right\},$$

where $\bar{x}_j^3(t)$ denotes the "best" company strength decision obtainable in period t using the backward MPH recursion, and $y_j^2(t)$ and $d_j^2(t)$ are the training cycle and training company deactivation decisions from Phase II.

The heuristics have been implemented in a *Decision Support System for Army Basic Combat Training* ***R****esource* *Management per Year*, or ***ARMY*** (see [8]). It is believed that the DSS can be extended to other Army training programs, and to training programs of other branches of military service (Navy, Air Force, Marines) as well.

4. Results

Comparison of Heuristic Schedulers

Three heuristic methods: (1) Heuristics-Used-In-Practice (HUIP); (2) Single-Pass Heuristic (SPH); and (3) Multi-Pass Heuristic (MPH), are evaluated using four performance measures: (1) CPU time; (2) objective function values; (3) resource utilization; and (4) training costs, for 12 test scenarios.

Results show that the SPH finds schedules 6.4 times faster than the HUIP method, and 1.7 times faster than the MPH if the time to find an initial feasible schedule (via SPH) is added to MPH processing time. The MPH is 3.7 times faster than the HUIP method.

Figure 2 compares instructor-to-student ratios for each heuristic procedure, where the *utopian* value of the performance measure is 0.64 (based on 96 periods). On average, the quality of the MPH and SPH schedules were 19% and 18.7% better than the HUIP schedules, respectively. For the scenarios considered, The HUIP, SPH, and MPH methods generate solutions that are (on average) approximately 74%, 87.5%, and 87.8% of the *utopian* value of the "quality" performance measure, respectively.

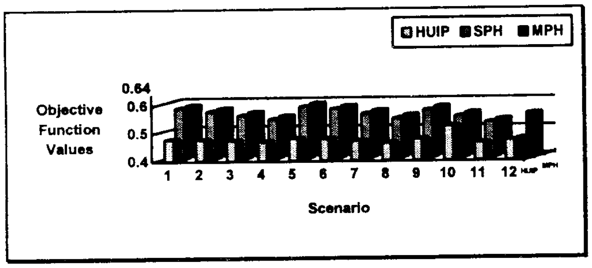

Figure 2. Comparison of Objective Function Values

Comparison of Optimal versus Heuristic Results

The high dimensionality of the real-world problem precludes implementation of an exact solution method which could be used as a yardstick to measure the effectiveness of the heuristic scheduler. Table 1 shows state space increases for incremental increases in time lags (based on a company strength step size of 5).

Lag	$x(t-l)$	$I(t)$	$x(t)$	State Space
1	21	21	21	9,261
2	21^2	24	21	222,264
3	21^3	27	21	$5.2\text{x}10^6$
4	21^4	30	21	$1.2\text{x}10^8$
5	21^5	33	21	$2.8\text{x}10^9$
6	21^6	36	21	$6.5\text{x}10^{10}$
7	21^7	39	21	$1.5\text{x}10^{12}$
8	21^8	42	21	$3.3\text{x}10^{13}$
9	21^9	45	21	$7.5\text{x}10^{14}$
10	21^{10}	48	21	$1.7\text{x}10^{16}$

Table 1. State Space for Increases in Time Lag

However, DP was implemented for a simplified one-period time lag, 48 period problem to compare DP versus heuristic results (see Figure 3). For the one-period time lag problem, the heuristic methods achieved results that were 91% of optimal for a small but representative set of test cases.

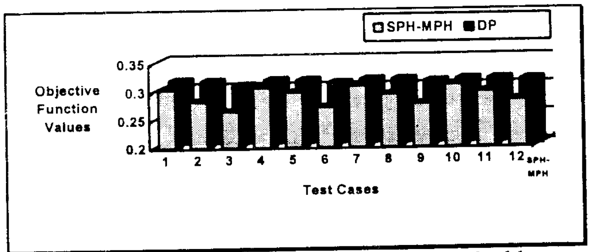

Figure 3. SPH-MPH vs DP for 1-Period Lag Problem

References

1. Afentakis, P., Gavish, B. and Karmarkar, U., 1984, Computationally Efficient Optimal Solutions to the Lot-Sizing Problem in Multistage Assembly Systems, *Management Science,* VOL. 30, pp. 222-239.
2. Bertsekas, D., 1987, *Dynamic Programming,* Prentice-Hall, Englewood Cliffs, NJ.
3. Billington, P., McClain, J., and Thomas, L., 1986, Heuristics for Multilevel Lot-Sizing with a Bottleneck, *Management Science,* VOL. 32, pp. 989-1006.
4. Blackburn, J. and Millen, R., 1982, Improved Heuristics for Multi-Stage Requirements Planning Systems, *Management Science,* VOL. 28, pp. 44-56.
5. Graves, S., 1981, Multi-Stage Lot-Sizing: An Iterative Procedure, *Management Science,* VOL. 16, pp. 95-110.
6. Manne, A., 1958, Programming of Economic Lot Sizes, *Management Science,* VOL. 4, pp. 115-135.
7. McGinnis, M., 1989, *Training Base Management Model User's Guide,* Version 2.0, U.S. Army Training and Doctrine Command Headquarters, Fort Monroe, VA.
8. McGinnis, M., 1994, *Resource Scheduling for the United States Army's Basic Combat Training Program.* Ph.D. dissertation, The University of Arizona.
9. Rao, Poornachandra P., 1990, A Dynamic Programming Approach to Determine Optimal Manpower Recruitment Policies, *Operations Research,* VOL. 41, pp. 983-988.
10. Wagner, M. and Whitin, T., 1958, Dynamic Version of the Economic Lot Size Model, *Management Science,* VOL. 5, pp. 89-96.
11. Yang, T. and Ignizio, J., 1987, Algorithm for the Scheduling of Army Battalion Exercises, *Computers and Operations Research,* VOL. 14, pp. 479-491.

Acknowledgements: E.F.G. research was partially supported by The Engineering Foundation, under grant RI-A-93-10.