

A Dynamic Programming Model for the Initial Entry Training Program of the United States Army

Lieutenant Colonel Michael L. McGinnis

The Operations Research Center, U.S. Military Academy, West Point, NY 10996
Email: fm0768@usma.usma.edu

Emmanuel Fernández-Gaucherand

Systems and Industrial Engineering Department, The University of Arizona, Tucson, AZ 85721
Email: emmanuel@sie.arizona.edu

1. Introduction

Each year, the US Army recruits and trains thousands of soldiers to fill vacancies in Army organizations. Installations responsible for training new recruits are located 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). Currently, manual heuristic methods, referred to as heuristics-used-in-practice (HUIP), are used to schedule basic 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. In this paper we report our work on: (1) a mathematical dynamic model of BCT and (2) a decision model for optimally scheduling basic training companies based on dynamic programming (DP). The reader is referred to McGinnis [2] for details, additional results, and references to previous work tangentially related to our approach.

2. Model Formulation

Mathematical Notation

j : year of the planning horizon, $j \in \{1, 2, \dots, J\}$. t : week of a given year j , $t \in \{1, 2, \dots, T_j\}$ 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. 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 (DA). $r_j(t)$: 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 . $y_j(t)$: training cycle length for companies starting in week t of year j . $I_j(t)$: number of idle training companies at the beginning of week t of year j .

Modeling Constraints

$$\underline{x} \leq x_j(t) \leq \bar{X}: \text{company strength constraint;} \quad (1)$$

$$\underline{y} \leq y_j(t) \leq \bar{Y}: \text{training cycle length constraint;} \quad (2)$$

$$0 \leq I_j(t) \leq \bar{I} \quad \forall (t, j): \text{problem feasibility constraint;} \quad (3)$$

$$r_j(t) = \delta(t) p(t) R_j: \text{expected number of recruits} \quad (4)$$

0-7803-1968-0/94\$4.00©1994 IEEE

to arrive for training in week t of year j , where R_j , determined by DA, is greater than the number of soldiers required to meet the needs of the Army in a given year j .

Stages are specified by week t and year j . We seek an optimal training resource schedule that maximizes the "quality" of training as measured by the *instructor-to-student ratio*. Assume one instructor per training company without loss of generality. 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 $J_\pi = \sum_{t=1}^{T_j-1} \frac{1}{x_j(t)}$. The optimal

sequence of decisions π^* is the one that maximizes the objective function $J_{\pi^*} = \max_{\pi \in \Pi} J_\pi$ for a fixed initial state.

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 constraints (1) and (2). The subsets of feasible

decisions to take at each stage t are $X_j[t, I_j(t)] \subset \Omega$ and

$$\mathcal{Y}_j[t, I_j(t)] \subset \Lambda.$$

Decisions, Scheduling Policy and Objective Function

Company strength $x_j(t)$ and training cycle length $y_j(t)$ decisions are made at the beginning of period t , $t = 1, 2, \dots, T_j - 1$, for all training companies that begin training that period. A sequence of such decisions is denoted

$$\pi = \{x_1(1), x_1(2), \dots, x_J(T-1); y_1(1), y_1(2), \dots, y_J(T-1)\}.$$

The set of feasible decisions satisfying constraints (1) through (4) above is denoted by Π .

State Transition Equation

The state of the system evolves according to the following balance equation for idle training companies:

$$I_j(t+1) = I_j(t) + \sum_{l \in L} \frac{r_j(t-l)}{x_j(t-l)} - \frac{r_j(t)}{x_j(t)}. \quad (5)$$

$\frac{r_j(t)}{x_j(t)}$ is an estimate of the number of companies to begin

training in week t of year j . $\sum_{l \in L} \frac{r_j(t-l)}{x_j(t-l)}$ 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 l weeks earlier. The possible values for $l \in L$ are contained within the set $L \in \{(10), (10,9), (10,9,8)\}$ which represent the possible time lags (in weeks) between the start and the end of training cycles (see [2] for details). The integer value of $I_j(t+1)$ in (5) is maintained by rounding the ratio terms of (5) according to rules used in practice as explained in [2].

3. Optimal Decision Models

Company strength and cycle length decisions depend upon past information that cannot be summarized in $I_j(t+1)$ alone. Additional information is made available through *state augmentation* [1]. 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 minimum number of variables necessary to make a decision in each period.

The augmented state space for a single period of the basic training problem (for 1988 training data) requires enumeration of (1) the number of idle training companies $I_j(t+1)$: 130; (2) company strength values $x_j(t-1), \dots, x_j(t-9)$ for the augmented state: $(101)^9$; and (3) 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 approximately 4.27×10^{20} possible states. Although DP 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 remains quite large. Table 1 shows the combinatorial explosion of the state space that occurs when attempting to obtain an optimal solution to the real-world basic training problem using DP for incremental increases in time lags (based on a company strength step size of 5, and assuming that the training cycle length is fixed at 10 weeks thus creating a nine-period time lag).

Lag	$x(t-l)$	$I(t)$	$x(t)$	State
1	21	21	21	9,261
2	21^2	24	21	222,264
3	21^3	27	21	5.2×10^6
10	21^{10}	48	21	1.7×10^{16}

Table 1. State Space for Increases in Time Lag

Other exact methods (e.g., integer and mixed integer programming, and complete enumeration) suffer from similar problems; see [2]. Thus far, the high dimensionality of the real-world problem has precluded the implementation of an exact solution method which could be used as a yardstick to measure

the effectiveness of the HUIP. These motivated the development of efficient heuristics [2].

4. Heuristic Decision Model

The heuristic procedures of [2] consist of two heuristics applied in three phases. 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. Experiments have shown that it is possible to improve resource schedules obtained via SPH. This observation led to the development and implementation of a *multi-pass heuristic* (MPH). The MPH makes additional passes through the planning horizon using a modified policy improvement step to further decrease training company strengths until no further improvements are possible.

The SPH and MPH methods have been implemented in a *Decision Support System for Army Basic Combat Training Resource Management per Year*, or **ARMY** [2]. Figure 1 illustrates the modular structure of the decision support system (DSS).

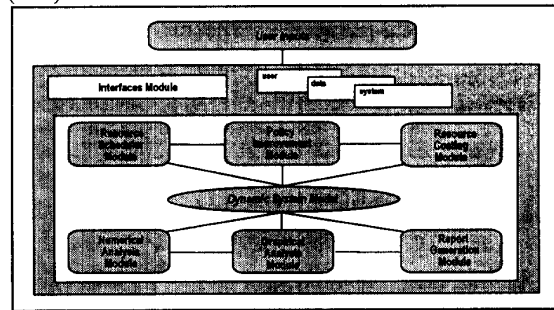


Figure 1. DSS Architecture and Modules

5. Optimal versus Heuristic Scheduling Results

Although the real-world problem is beyond the scope of optimal solution methods, DP was implemented for a simplified one-period time lag, 48-period problem to compare DP versus heuristic results (see Figure 2). On average the SPH-MPH heuristic methods achieved results were 91% of optimal; a significant improvement over the heuristics-used-in-practice.

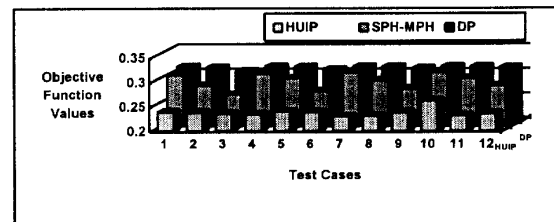


Figure 2. Scheduling Results for a 1-Period Lag Problem

6. References

- [1] D. Bertsekas, *Dynamic Programming*, Prentice-Hall, Englewood Cliffs, NJ, 1987.
- [2] M. McGinnis, *Resource Scheduling for the United States Army's Basic Combat Training Program*. Ph.D. dissertation, The University of Arizona, 1994.