TraDE: Training Device Selection via Multi-Objective Optimization

Slawomir Wesolkowski, Nevena Francetić, Stuart C. Grant

***Abstract*— Training planning is a recurring military problem. Since training programs can utilize multiple training devices with varying costs and training capabilities, selecting the types of devices required is a complex trade-off problem. Furthermore, the placement of these devices is critical due to the time and costs involved in travelling to and from the location of a training device. In this paper, we introduce a device bin-packing-and-location-based model, Training Device Estimation (TraDE), to study the computation of heterogeneous device mixes including the location of each device with respect to numerous objectives including various costs and training time. We apply the multi-objective Non-dominating Sorting Genetic Algorithm II to the TraDE model on a population represented by two-dimensional chromosomes. Finally, we also present a new mutation type to handle the nonlinearity inherent in a dual optimization problem which includes scheduling and location optimization. We clearly show that the new mutation operator produces superior results to the standard mutation operator.**

I. Introduction

EVERY soldier has to successfully complete mandatory military training. The training usually consists of successfully completing a series of tasks. Moreover, the same equipment can be used for training of various tasks, but the device's capacity, efficiency and costs per use are task-dependent. Even when considered at a single location, there is a need to match a device type used for a training task for each soldier, as there are many compatible devices. However, military bases are usually spread throughout a country and every location does not have or need the same training equipment. Moreover, not all training devices can be placed at every base due to terrain, weather or land use limitations. Hence, individual soldiers or parts of a unit may be required to travel to another location in order to obtain training. In this complex set of constraints, we address the question of what is the minimum number of training devices at each admissible location which are necessary to ensure that all soldiers are able to complete required training. There are multiple objectives to consider such as minimizing capital and operating costs of the devices, as well as minimizing the soldiers' total travel costs and their total time spent in training.

Given the wide variety of training tasks and the large number of possible training devices, finding a good device mix is difficult [2]. There are many key parameters to take into account. For example, the triangle model [3] considers device fidelity, training delivery method, and training content to determine good device mixes. On the other hand, the FAPV approach analyzes the stage of the trainee's learning process to select an effective training environment [6]. Furthermore, the Army Training [11] and the Stochastic Fleet Estimation (SaFE) [15] models approach the problem from a resource allocation standpoint. The Army Training Mixed Model [11] considers a single cost objective training device mix problem for one location. On the other hand, SaFE [15] is applicable to a multi-objective fleet mix allocation problem also at a single location. Like in the SaFE model, the problem of selecting training devices is similar to a sparse-multi-capacity 2D bin packing application for job scheduling (the device dimensions are soldier capacity and training duration). However, since the number of each device type used is not directly proportional to other training device types, methods normally applied to multi-capacity bin packing problems are not applicable [15].

In this paper, we also consider the training problem from a resource allocation point of view. First, we must determine the minimum number of devices of each type necessary for all soldiers to be able to complete their full required training. Second, we need to determine where to locate these devices to minimize the travel costs of all soldiers. Both problems, for a large number of locations and devices, are usually individually difficult to solve. Determining only a device mix for each task is comparable to determining a fleet configuration, which is based on bin packing and scheduling, two NP-complete problems [1], [16]. Moreover, given that a change in the location and number of any training devices would significantly (and nonlinearly) affect the computed objective functions, performing these two optimizations sequentially would not ensure an overall optimal solution. Therefore, we propose to carry out a joint optimization of the number of devices and their locations. For our model, called the Training Device Estimation (TraDE), the joint optimization is represented by a multi-dimensional chromosome for a population member in the genetic algorithm used to explore the solution space. Training tasks correspond to genes in a chromosome, and every gene carries the information about the device configuration for the given task as well as the locations of the devices.

We evaluate the solutions with respect to four objectives: acquisition and operating costs of the devices, as well as total travel costs for all soldiers and total time spent in training. Multi-objective optimization allows us to evaluate the solutions without the need to estimate weights of objectives with respect to each other [4]. Furthermore, we are exploring a vast solution space. Indeed, the training consists of multiple tasks; for each task, there is a set of devices which are equally compatible for use in training; a device of each type can be placed at a number of locations; and multiple devices of various types can be designated for training of a single

Slawormir Wesolkowski, Nevena Francetić and Stuart C. Grant are with the Defence Research and Development Canada, Ottawa, Canada (email: s.wesolkowski@ieee.org).

DRDC-RDDC-2014-P7

task (this depends on device capacity and number of soldiers needing to be trained). Hence, we explore a large number of combinations using the Non-dominating Sorting Genetic Algorithm II (NSGA-II) [4].

The paper is organized in the following manner. We give a detailed problem description in Section II. Section III gives the TraDE model specifications. In Section IV, we compare the two mutation types and provide an overview of the results. Section V contains some concluding remarks.

II. Model Description

A. Tasks and Devices

Building a training system to deliver small arms training entails decisions regarding many system components, such as personnel, infrastructure, funding, and equipment. The main goal of training is to ensure that all soldiers obtain satisfactory sensory-motor skills, such as short release and follow-through, and cognitive skills, like the coordination of small arms fire amongst dispersed groups. For example, new recruits might acquire basic marksmanship skills, which is one of the training tasks performed during a three month basic training program [7].

The devices used for training can greatly vary. Traditionally, the training was performed using unloaded weapons and live fire. Today, the devices can vary from a shooting range to a virtual simulator or an online computer course [5]. Depending on their type, the devices vary in their capabilities, capacities, but also in costs, risks and infrastructure requirements. No device can be used for training of all tasks. For example, a computer cannot be utilized for training of body positioning and rifle holding for a stable and comfortable firing. Similarly, training with non-lethal ammunition on a live fire range is not adequate for training long range engagements due to ballistic differences in non-lethal and operational ammunition [7]. However, a task is often trainable by different device types, and devices have versatile functionality, some of which may overlap. Hence, different task-to-device assignments are studied to find a better combination with respect to our objectives.

In TraDE, to carry out soldier training, a fixed set of training *tasks*, such as quick aim shooting or safe handling are specified [5]. We denote by T the total number of tasks. All tasks must be completed by all soldiers. Each task is characterized by a set of compatible *devices* that can be used to complete the task. There are D device types. It is assumed that all compatible device types for a task are used to achieve the same level of proficiency in that task[1]. The following two values characterize the task-device type dependency.

- A device type training effectiveness is measured in $\tau(t, i)$, the time required for the task training t using a device of type i.
- Device types vary in the cost, $c(t, i)$, incurred when completing a training task t on the device of type i.

[1]We suspect that this is most likely not a correct assumption and a model that will take this information into account is currently being developed.

In TraDE, each device type i, $i = 1, 2, \ldots, D$, is characterized by the following five values:

- the soldier capacity sc_i,
- the total amount of time a device of this type can be used per year, θ_i,
- the acquisition cost $\mathrm{A_i}$,
- the annual operating cost $\mathrm{Op_i}$, and
- the set of possible locations where a device of this type can be placed, $\mathcal{L}_i$.

Soldiers needing training are stationed at various *locations* (e.g., bases and other facilities) across the country. When training is required, the unit (or a part of it) may have to travel to the training location, complete the training, and return to their home location. Since some tasks can take multiple days to train, each soldier incurs costs for staying at the training location for the duration of the training task. Also, there are travelling costs between the home location and device locations per soldier per task.

The training program, which specifies the set of tasks to be completed by all soldiers, device types' characteristics, as well as the number of soldiers needing training and their location are pre-determined and fixed. Data used to study the TraDE model has been simulated. Although this data was obtained with the help of subject matter experts, it should not be construed as to represent in any way the capabilities or deficiencies of the Canadian Armed Forces.

B. Device-to-Task Assignment Configuration

As described in Subsection II-A, various device types can be used for task training. Thus, depending on the total number of soldiers requiring training and the capacities of device types, a training task can be distributed over several compatible devices (i.e., not necessarily of the same type), depending on the time limit in which the training has to be completed. This configuration problem is equivalent to determining a configuration of aircraft to transport a large amount of cargo: depending on the capacity and speed of each plane type, the transportation can be distributed over various combinations of aircraft, depending on the size of the cargo and the time in which the task has to be performed [15].

More precisely, for a task t, a device *configuration* is a D-tuple $(n_{t_1}, n_{t_2}, \ldots, n_{t_D})$ where n_{t_i} is the number of devices of type i (in this study, n_{t_i} is not constrained to a particular value; however, it is still implicitly limited by the combination of training tasks and soldiers who need to complete those tasks). Only considering the soldier capacities for each device type and the number of soldiers who need to complete training, we determine all possible minimum device configurations adequate for completing a training task by all soldiers. That is, for each task t, we pick the non-dominated set of configurations among all which satisfy the following:

$$\text{total number of soldiers} \leq \sum_{i=1}^{D} n_{t_i} sc_i,$$

and the total time spent in training of all soldiers on this task using this configuration does not exceed the predefined limit.

The set of all device configurations for each task is the *configuration set*. The configuration set is determined prior to the execution of the model, and allows us to fully specify the search space (even if it is still huge) of admissible configurations for each task during model execution. The number of admissible configurations for a task can vary from only one if the task can be performed on a single device, to a mix of devices of one or more types. Hence, the combinatorial complexity of the problem depends on the number of tasks and the number of device configurations for each task [10].

C. Device Number and Location Determination

The goal of the present study is to determine a *device mix per location*; i.e., the number of devices of each type required at a given location while minimizing the capital, operating and travel costs together with the total time spent in training. In TraDE, a solution has a device configuration assigned to each training task, giving a number of devices of each type required for the successful completion of the training task by all soldiers. Moreover, the model assigns a location for each device in the configuration, which may, or may not, coincide with the location of sets of soldiers to be trained using the device. More precisely, a solution in TraDE has the following information for *every training task*:

$$(n_1, n_2, \ldots, n_D) \times ((l_{11}, l_{12}, \ldots, l_{1n_1}), (l_{21}, l_{22}, \ldots, l_{2n_2}), \ldots, (l_{n_D 1}, l_{n_D 2}, \ldots, l_{n_D n_D})), \quad (1)$$

where n_i is the number of devices of type i necessary to complete the given task in this configuration, and l_{ij} is a location of the j^{th} device of type i. Note that even though this information corresponds to only one task t, we omitted the subscript t to improve the readability.

To achieve our goal of determining the device mix at each location, we adapt the bin-packing method applied in SaFE [15]. First we compute the total time over all tasks for which a device type is used in the solution. That is, given a location l and a device type i, denote by $n_{t_i}(l)$ the number of devices of type i which are assigned to the location l for the task t by the solution. Then, the total time the device type i is utilized at the location l in training is:

$$T_i(l) = \sum_{t=1}^{T} n_{t_i}(l) \cdot \tau(t, i).$$

Now, given our restrictions on annual device usage θ_i, we compute

$$f_i(l) = \left\lceil \frac{T_i(l)}{\theta_i} \right\rceil,$$

the number of devices of type i which are required by the solution at the location l, that is, the device mix at each location.

Note that the number of devices in a device configuration for a training task depends on the number of soldiers to be trained. Since the number of soldiers is constant, the number of devices of each type in a configuration is limited from above. Therefore, even though there are no constraints on the maximum number of devices at each location (which technically can be infinite), the model keeps these numbers always below certain maximum dependent on the number or soldiers and the number of training tasks.

In TraDE, the set of all soldiers is ordered prior to the model execution. The device configuration specifies the number of devices of each type necessary for the training of all soldiers for a given task (see equation (1)). First we decide in which order we will match device types to soldiers. The order is determined by sorting the D-tuple $(n_1, n_2, \ldots, n_D)$ from the smallest to the largest number. Given that $n_{i_0} = min_i\{n_i : i = 1, 2, \ldots, D\}$, then the devices of type i_0 are considered first. The device type i_0 has capacity sc_{i_0}, and we match the soldiers to the device locations in the following order, starting from the first soldier:

Location	Soldiers
$l_{i_0 1}$	1 to sc_{i_0}
$l_{i_0 2}$	$sc_{i_0} + 1$ to $2\, sc_{i_0}$
$\vdots$	$\vdots$
$l_{i_0 n_{i_0}}$	$(n_{i_0} - 1)\, sc_{i_0} + 1$ to $n_{i_0}\, sc_{i_0}$

The remaining soldiers are then matched to devices of other types in the same manner. Given the home location of each soldier, we can determine the travel and other expenses incurred by the soldier completing the training for an task at the location specified by the given solution.

III. Multi-objective Optimization

Since we do not want to estimate weights of each objective with respect to each other prior to applying an optimization algorithm, we need to use a multi-objective genetic algorithm. TraDE uses the NGSA-II [4] algorithm to obtain an approximation of the Pareto Front with respect to many objectives (in this case four). Moreover, using a multi-objective algorithm allows us to study the trade-offs between different objectives and how these objectives affect the device mix at each location.

In Subsection III-A, we discuss the structure of the chromosomes of the population members. Like every genetic algorithm, NGSA-II applies the crossover and mutation operators (see Subsection III-B) on the current set of solutions, called the parents, to obtain the new set of solutions, called the children. The parent and children population are combined and re-evaluated, and the fittest members of the population are taken to be the parents in the next generation.

A. Solution Representation

In TraDE, every solution is represented by a two-part chromosome. A chromosome consists of genes, and a gene corresponds to a training task'seeds device configuration with locations, as described by equation (1).

Two-dimensional chromosomes have been considered in joint problem optimization [9] and in two-dimensional packing [8]. However, for these problems the chromosomes had a fixed size in both dimensions. In our problem, the number of required device locations depends on the number of devices in the configuration. Hence the length of the array of locations varies with the device type. Having a two-part chromosome requires adapting the crossover and mutation operators in a genetic algorithm [8], [9]. In TraDE, the standard crossover operator can be used. On the other hand, we study two variations of the mutation operator. The mutation operators differ by which part of a gene is altered and how this is done. More details are provided in Subsection III-B.

B. Crossover and Mutation

The crossover and mutation operators act on the genes of a chromosome. In TraDE, genes correspond to the training task configuration. We use a standard crossover operator which picks, with equal probability, from one of the two parents the device configuration and locations for the given task. However, not all members of the population are equally fit to be parents to the next generation. Population members are sorted in non-dominated fronts. A member in front i has a probability of $\frac{1}{i}$ of being picked as a parent for the next generation. In this way, members that are more likely to produce better solutions are picked with higher probability to be parents, which speeds convergence for a small number of objectives [12].

Given that each solution has two-part chromosome, the mutation operator can be defined in a number of ways. We consider two types:

- **Mutation Type 1:** In this variation, we have only one mutation rate, μ, representing the probability to change a task configuration of a child. If the task configuration is being replaced, we pick uniformly at random an admissible device configuration from the configuration set which gives us the first part of the gene (see equation (1)). Then for each device type i, we assign uniformly at random n_i locations from the set of all eligible locations $\mathcal{L}_i$, to obtain the second part of the gene.
- **Mutation Type 2:** The mutation rate is a three-tuple (μ_1, μ_2, μ_3). A task configuration of a child is mutated with probability μ_1. If the task configuration is being changed, we have a possibility of altering the device configuration or keeping the configuration and only mutating device locations. The device configuration is replaced with probability μ_2. In this case, another device configuration for the task is picked uniformly at random from the configuration set. However, we keep as many device locations unchanged as possible. That is, if the original task configuration is given as in equation (1), and the new device configuration is $(N_1, N_2, \ldots, N_D)$, then for every $i = 1, 2, \ldots, D$, let $m_i = \min\{n_i, N_i\}$, and assign locations $\left(l_{i1}, l_{i2}, \ldots, l_{im_i}, L_{i(m_i+1)}, L_{i(m_i+2)}, \ldots, L_{iN_i}\right)$ for the devices of type i, where locations L_{ij} are picked uniformly at random from the set $\mathcal{L}_i$ ($j = m_i + 1, m_i + 2, \ldots, N_i$).
  Otherwise, the device configuration remains the same. In this case, we go through the list of device locations, and replace the location l_{ij} with probability μ_3 by a randomly chosen location from $\mathcal{L}_i$.

We tested the TraDE model with both mutation types. We discus the results of these experiments in Section IV.

C. Objective Functions

Before we define the objective functions, we need to introduce some notation. We denote by $\mathcal{S}$ the set of all soldiers who need training. Also, we denote by x a specific solutions, that is a device configuration and their location specified for all tasks.

The fitness of each population member is evaluated with respect to the following four objective functions which are to be minimized.

- *Acquisition cost* :
  $$\sum_{i=1}^{D}\Big(\sum_{l\in\mathcal{L}_i} \max(0, (f_i(l) - \mathrm{existing_i(l)}\,))\Big) \cdot \mathrm{A_i},$$
  where $\mathrm{existing_i(l)}$ is the number of devices of type i which already exist at location l.
- *Operating cost*:
  $$\sum_{i=1}^{D}\Big(\sum_{l\in\mathcal{L}_i} f_i(l)\Big) \cdot \mathrm{Op_i} + \sum_{\mathrm{s}\in\mathcal{S}} \mathrm{U_x(s)},$$
  where $U_x(s)$ is the total cost incurred by soldiers to complete the full training on devices as specified by solution x.
- *Travel cost*:
  $$\sum_{s\in\mathcal{S}} V_x(s),$$
  where $V_x(s)$ is the total cost incurred by soldier s for a return trip between his/her home base and the device location, as specified by solution x, to perform a training task, summed over all tasks. In addition, $V_x(s)$ contains the costs of soldier s staying at the location of the device for the duration of the training task, summed over all tasks.
- *Training time*:
  $$\sum_{s\in\mathcal{S}} T_x(s),$$
  where $T_x(s)$ is the total time in takes a soldier s to complete all tasks on the devices specified by solution x.

A device location is necessary to determine the expense of a return trip for each soldier from the home location to the device location, as well as for calculating the costs incurred by the soldier staying at the device location for the duration of the training. To simplify the model, these computations are done per task, with an assumption that after completing the training task, the soldier always returns back to the home location before starting a new training task.

D. Optimization Initialization

The algorithm initializes the first population randomly. For the device configuration part of the chromosome, a random admissible configuration is chosen from the predetermined configuration set for each training task. For the device location part, eligible locations are randomly assigned to the devices for the previously chosen configuration.

After initialization, some of the randomly chosen members of the first population are replaced by *seeds*, i.e. approximate best-fit solutions, in order to bootstrap the genetic algorithm. A seeding mechanism has been implemented in order to generate an optimal or nearly optimal solution with respect to each objective separately. In particular, given a configuration for a task from the configuration set, the value of all objectives, except the travel cost, can be computed. The travel cost depends on the locations of the devices which are not specified by device configurations. To get a value for the travel cost objective, for each device configuration, a location is assigned for one device at a time, such that the incurred travel cost by this device placement is minimized. Finally, for each task, the configuration from the configuration set with the minimum value for the given objective is taken, to obtain a seed.

A planned set of devices could also be added as a seed. This would allow an analyst to determine whether a planned solution would be part of the non-dominated front or might be superseded by solutions on the non-dominated front.

IV. Results

The experiments were performed on a data set which has the following parameters: 8,550 soldiers that need training at 13 locations, four distinct training devices, and 11 tasks which each soldier has to complete. For the current data set, the training devices can be positioned at 22 different locations. At nine of those locations, there are no soldiers. We assume an inter-location travel cost between $0 (no travel required) and $2000 with different per day costs at a device's location (in the $100-$500 range).

We performed 30 experiments, that is, 15 experiments for each mutation type. Every experiment had the number of population members set to 200, the number of generations equal to 10,000, producing $15 \cdot 200 = 3,000$ solutions in total. We implemented the TraDE model in MATLAB and ran it on a Windows 7 PC with 3 GHz twelve core CPU and 18 GB RAM. Each experiment took approximately 12 hours to complete. The experiments with Mutation Type 1 had $\mu = 0.1$. The experiments with Mutation Type 2 had $(\mu_1, \mu_2, \mu_3) = (0.1, 0.5, 0.1)$. Subject matter experts helped in devising the input data set.

The results of the 15 experiments with a specified mutation type were combined and re-evaluated to produce a *combined* non-dominated front (combined NDF). We obtained two combined NDFs, corresponding to the two mutation types, having 1,388 and 1,480 population members. Then the sets of solutions in the two combined non-dominated fronts were united and resorted, to produce a *super* non-dominated front (super NDF) having 2,002 solutions. Few of the solutions in the super front were achieving the same values in all four objective functions; hence, the repeated solutions were eliminated. Ultimately, we obtained 1,990 non-dominated solutions in the super front.

The results of the experiments are compared in the following way. For a set of non-dominated solutions A, define $\mathcal{O}(A)$ to be the multi-set of all tuples of the objective function values of the solutions in A. Assume that the union of the set A with another set of solutions was re-evaluated to produce a combined non-dominated front of solutions B. We compare the 4-tuples of the objective function values of the solutions in A with the 4-tuples of the objective values of the solutions in B by computing the portion of A covered by B:

$$P = \frac{|\mathcal{O}(A) \cap \mathcal{O}(B)|}{|\mathcal{O}(A)|}.$$

We say that $P \cdot 100\%$ of A is covered by B.

Recall, we performed 15 experiments for each mutation type. Hence, we can compute the percentage of the non-dominated front of each experiment covered by the combined non-dominated front, or alternatively by the super front. In Table I, we present only the mean value and standard deviation of these percentages over all 15 experiments.

TABLE I

Comparison of coverage of NDFs for the two mutation types

	Mutation Type 1	Mutation Type 2
Size of Combined NDF	1,388 (out of 3,000)	1,480 (out of 3,000)
Mean and standard deviation of the coverage of each experiment's NDF by their respective combined NDFs	0.4627 ± 0.0494	0.4933 ± 0.1319
Mean and standard deviation of the coverage of each experiment's NDF by the super NDF	0.2100 ± 0.0489	0.4573 ± 0.1393

We found that, on average, about one half of the non-dominated front of an experiment is covered by the combined non-dominated front in both cases. Moreover, a t-test fails to reject the hypothesis that the coverage of the NDFs from the experiments by the respective combined fronts is determined by a normal distribution with the respective observed mean at the 5% significance level (with p-value$= 1$).

However, the last row of Table I shows that the two mutation types substantially differ when the non-dominated fronts from the individual experiments are compared with the super non-dominated front. The mean value of the coverage of the non-dominated front from an experiment with Mutation Type 1 is considerably lower than the mean value of the coverage of the non-dominated front from a trial with Mutation Type 2. In addition, a paired t-test, with the assumption of unequal standard deviations, fails to reject the hypothesis that the mean of the coverage of the NDFs from the experiments with Mutation Type 1 is smaller than the mean of the coverage of the NDFs from the experiments with Mutation Type 2, at the significance level of 5% (with p- value$= 0.99$). This trend is clearly reflected on the coverage of the combined non-dominated fronts for each mutation type by the super non-dominated front. As shown in Figure 1, only 45% of the combined NDF with Mutation Type 1 is covered by the super front. On the other hand, 93% of the combined NDF with Mutation Type 2 is contained in the super non-dominated front.

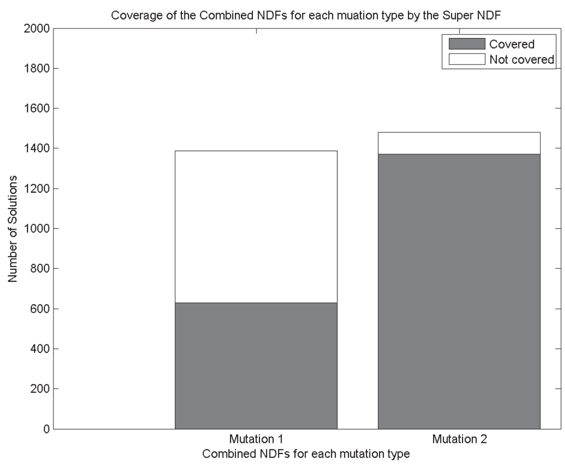

Fig. 1. Coverage of the combined mutation type-specific non-dominated fronts by the super front

A. Device Mixes

In this section, we look into some traits and general characteristics of the produced device mixes.

For example, in our data, Devices A and B have similar functionality; the sets of tasks that can be performed on them almost coincide. Moreover, these devices are versatile, that is, almost all training tasks can be performed on them. The correlation coefficient between Devices of type A and B in the solutions from the super front is -0.76. This inverse relationship could have been expected given the compatibility of the two devices. Recall that the TraDE model keeps the number of devices capped and therefore if one device is used more some other device must be used less. However, even though these devices have similar functionality, their training time per task, as well as cost per use for a task differ; hence, they cannot completely replace one another.

However, examining the solution set, we see that Device type B is favoured. Indeed, Table II shows the average of the number of devices of each type taken over all solutions in the super non-dominated front (summed over all locations) required to perform the training of all soldiers. Examining the input values to the model, we can see that Device B is more efficient and less costly than Device A; therefore, the obtained solution set confirms our expectations. Furthermore, on average, Devices of type B and D are the most used. Figure 2 depicts the number of hours devices of a given type are used for a training task, averaged over all solutions. While Device B is versatile and used for the majority of tasks, Device D is primarily used for tasks which cannot be trained on Devices of type B.

TABLE II

THE AVERAGE NUMBER OF DEVICES OF EACH TYPE REQUIRED PER SOLUTION IN THE SUPER NDF.

Device A	Device B	Device C	Device D
6.5754	51.9101	3.2612	26.9136

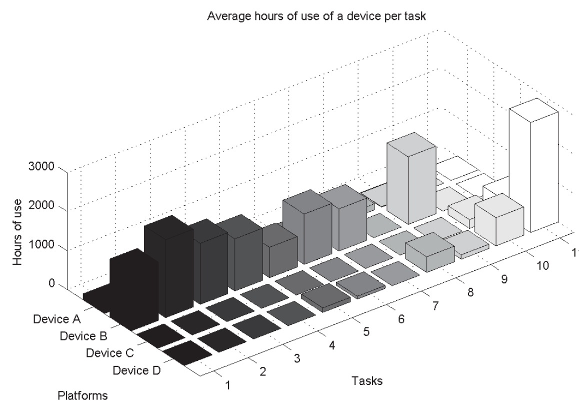

Fig. 2. Average of hours a device type is used in training per solution in the super NDF.

Figure 2 identifies the average task-dependent device mix, suggesting which device types are more efficient for the given tasks, helping in planning future procurement. Moreover, studying the number of devices of each type per location, averaged over all solutions, gives us which locations may profit the most from the introduction of new training equipment.

Next we consider the solutions in our super NDF which have the minimum value for one of the objectives. Since we have a large super NDF, we obtained many distinct solutions reaching the minimum for each objective function. This many-to-one mapping is possible given that each solution is based on a different device-to-task assignment, i.e., different configurations. We present the solutions having the smallest value of the fixed objective function and are the closest to the minimum values of the other objectives. That is, denote by $\mathcal{X}$ the set of all solutions in the super non-dominated front. Let

$$m_j = \min_{x \in \mathcal{X}} f_j(x),$$

be the minimum value of the jth objective function over all solutions in the super NDF, and let

$$M_j = \{x \in \mathcal{X} \; : \; f_j(x) = m_j\},$$

be the set of all solution reaching the minimum in the jth objective. Below, we present solutions meeting the following minimum:

$$\min_{x \in M_j} \sum_{j=1}^{4} \left(1 - \frac{m_j}{f_j(x)}\right).$$

Recall a device mix specifies a number of devices of each type at all locations necessary for completing all training tasks for all soldiers in a given time period. Solutions which have capital cost equal to zero use up the existing devices and usually only a small subset of them. Moreover, no solution in the super NDF utilizes all of the existing devices, potentially indicating that some of the existing resources at certain locations are redundant (however, this result could also have been obtained due to erroneous input data provided by subject matter experts).

In terms of costs, the values of the operating cost and travel cost objective functions are of similar orders of magnitude. The solution with the smallest operating cost (see Fig. 3) greatly favours use of Device B at almost all locations. This is consistent with the fact that Device of type B is inexpensive. However, its capacity is considerably smaller than the capacity of competitive devices on which the same task can be performed (on average only 9.8% of the capacity of other devices). Moreover, the solution in Figure 3 has about 3.8 times higher travelling costs than the solution with the minimum value for travel costs (see Figure 4).

The solution in Figure 4 with the smallest value for the travel costs has approximately 1.5 times higher operating costs than the minimum, and interestingly, does not require devices to be present at all locations. However, it does emphasize the presence of devices at locations where the majority of soldiers are stationed. In our input data, Locations 1-5 are home bases for 75% of soldiers.

The device mix in Figure 5 corresponds to the solution with the smallest sum of all costs: capital, operational, and travel. It is a representation of a solution with objective functions' values between those in solutions given in Figures 3 and 4. In this solution, we have devices at all locations; however, most devices are still concentrated at Locations 1-5.

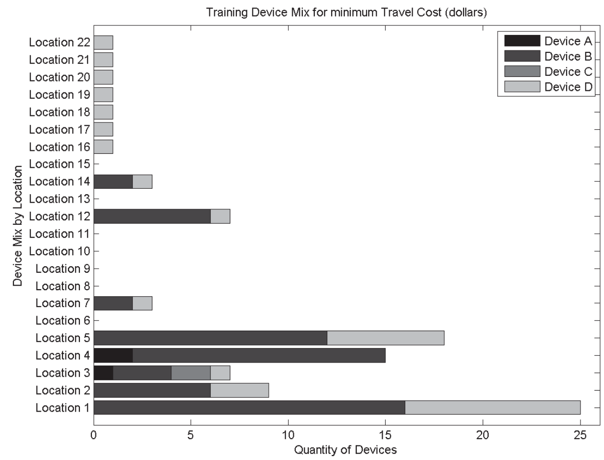

Fig. 4. A device mix for a solution with the minimum travel cost

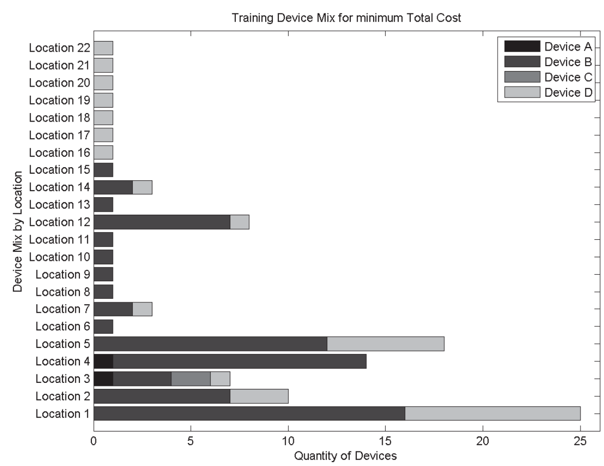

Fig. 5. A device mix for a solution with the minimum of sum of capital, operational, and travel costs

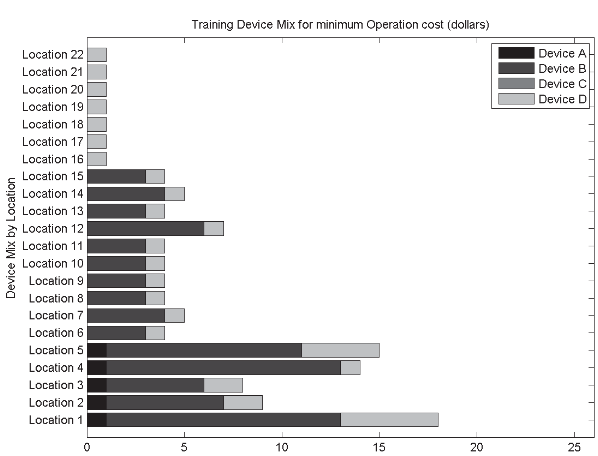

Fig. 3. A device mix for a solution with the minimum operational cost

Figure 6 gives a device mix for a solution with the minimum total training time among the solutions in the super NDF. The objective values for the operational and travel costs in this solution are slightly higher than in the solution with the minimum travel costs (see Figure 4), and the device mixes are almost identical. They only differ in the device mix at Locations 1 and 2. However, the correlation between the travel cost objective and the total training time objective among the solutions in the super NDF is only 0.44. Nonetheless, in many experiments, the solutions reaching minimum values in these objectives have consistently had identical or almost the same device mixes for our input data. One possible answer would be that, for our data, these two objectives result in similar optimal solution sets; however, the relationship between suboptimal solutions in these two objectives could be highly nonlinear. This is certainly not well understood and merits further study.

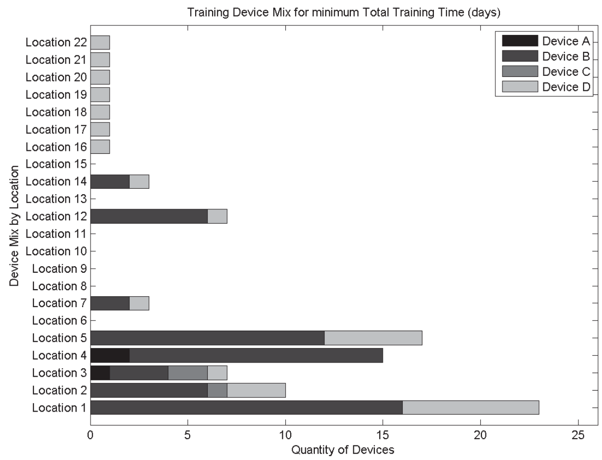

Fig. 6. A device mix for a solution with the minimum training time

V. Conclusion

Determining the number of training devices to accomplish training tasks is a complex problem, especially when one must consider the optimal location placement of the devices. TraDE gives a set of solutions which provide trade-offs between multiple objectives. Each solution gives an allocation of devices at various suggested locations. Furthermore, the solutions can be used to identify existing devices which are redundant and can be removed to lower annual maintenance costs. Also, solutions suggest which new devices may be placed at various locations for more efficient training. Moreover, we can infer which device types are more effective for meeting a given objective, and how many of them and where they should be placed.

The introduction of locations where devices should be placed is reflected in the double chromosomes for the members of the population in the genetic algorithm. In TraDE, the crossover operator is suggested naturally by the formulation of the problem. However, we tested two mutation types and found that Mutation Type 2 performs significantly better in terms of obtaining non-dominated solutions and overlap with the super NDF. This type of mutation operator set up may help in other complicated optimization problems where several NP-hard problems are being solved at the same time.

The next stage of work will involve running the algorithm for a longer number of generations to ensure further convergence. A more detailed analysis of the trade-offs between the various objectives will also be performed. Moreover, we plan to adapt the model to more complex data which may involve introducing task prerequisites, having compound tasks for which subtasks can be completed on different devices, and having different training programs. Finally, we will look into trade-offs in training effectiveness between different device types.

VI. Acknowledgments

Mr. Nathaniel Horvath, a student at the University of Waterloo, participated in the development of the initial version of the TraDE model when he was a co-op student at DRDC.

References

[1] S. Baker, A. Bender, H. A. Abbass, and R. Sarker, "A scenario-based evolutionary scheduling approach for assessing future supply chain fleet capabilities," in *Evolutionary Scheduling (studies in Computational Intelligence)*, P. Cowling, K. Tan, and K. Dahal, Eds. Berlin, Germany. Springer-Varlag, 2007.

[2] J. Cohn, "Building Virtual Environment Training Systems For Success," *Virtual Environments for Training and Education: Developments for the Military and Beyond*, Westport, Praeger Security International, pp. 193–207, 2008.

[3] M. Daly and D. Thorpe, "Balancing Simulated and Live Naval Fleet Training," *Interservice/Industry Training Simulation Education Conference*, Orlando, 2009.

[4] K. Deb, "Multi-Objective Optimization using Evolutionary Algorithms," Chichester: John Wiley & Sons, 2001.

[5] Department of National Defence. Canadian Forces Operational Shooting Programme, DND, Ottawa, Canada. 2007.

[6] G. Frank, R. Helms and D. Voor, "Determining the Right Mix of Live, Virtual and Constructive Training," *Interservice/Industry Training Simulation Education Conference*, Orlando, 2000.

[7] S. C. Grant and S. Wesolkowski, "A Multi-Objective Optimization Approach to Selecting Sets of Training Devices," in *Summer Simulation Multi-Conference*, Toronto, 2013.

[8] S. Jain and H. Gea, "Two-dimensional packing problems using genetic algorithms," *Engineering with Computers*, vol. 14(3), pp. 206-213, 1998.

[9] A. Khajavirad and J. Michalek and T. Simpson, "An efficient decomposed multiobjective genetic algorithm for solving the joint product platform selection and product family design problem with generalized commonality," *Structural and Multidisciplinary Optimization*, vol. 29(2), pp. 187- 201, 2009.

[10] M. Mazurek and S. Wesolkowski, "Fleet mix computation using evolutionary multiobjective optimization," *IEEE Symposium on Computational Intelligence in Miulti-Criteria Decision-Making 2009*, pp. 46–50, 2009.

[11] K. Murty, P. Djang, W. Butler and R. Laferriere, "The Army Training Mix Model," *Journal of the Operational Research Society*, no. 46, pp. 294 – 303, 1995.

[12] M. Wagner and T. Friedrich, "Efficient parent selection for Approximation-Guided Evolutionary multi-objective optimization," *IEEE Congress on Evolutionary Computation*, pp. 1846–1853, 2013.

[13] S. Wesolkowski, D. Wojtaszek, and K. Willick, "Multi-objective Optimization of the Fleet Mix Problem Using the SaFER Model," *IEEE CEC*, Brisbane, Austraila, 2012.

[14] D. Wojtaszek and S. Wesolkowski, "Multi-objective evolutionary optimization of a military air transportation fleet mix with the flexibility objective," *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 96–101, 2011.

[15] D. Wojtaszek and S. Wesolkowski, "Evaluating the Flexibility of Military Air Mobility Fleets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no.4, pp. 435–445, 2014.

[16] D. Wojtaszek and S. Wesolkowski, "Military Fleet Mix Computation and Analysis," *IEEE Computational Intelligence Magazine*, vol. 7, no. 3, pp. 53–61, 2012.