A graph coloring approach to the deployment scheduling and unit assignment problem

Mark Zais & Manuel Laguna

Journal of Scheduling

ISSN 1094-6136

J Sched
DOI 10.1007/s10951-015-0434-0

Springer

J Sched
DOI 10.1007/s10951-015-0434-0

A graph coloring approach to the deployment scheduling and unit assignment problem

Mark Zais[1] · Manuel Laguna[1]

Abstract We address one of the external factors of personnel inventory behavior, deployments. The configuration of persistent unit deployments has the ability to affect everything from individual perceptions of service palatability to operational effectiveness. There is little evidence to suggest any analytical underpinnings to the U.S. Army deployment scheduling and unit assignment patterns. This paper shows that the deployment scheduling and unit assignment problem can be formulated as an interval graph such that modifications to traditional graph coloring algorithms provide an efficient mechanism for dealing with multiple objectives.

Keywords Graph coloring · Interval graph · Optimization · Scheduling · Metaheuristics

1 Introduction

This paper proposes a graph coloring approach to the deployment scheduling and unit assignment (DSUA) problem and addresses one of the external factors of personnel inventory behavior, deployments. The configuration of persistent unit deployments affects both individual perceptions of service palatability and operational effectiveness. We focus on the unit deployment problem faced by the U.S. Army. There is little evidence to suggest any analytical underpinnings to the U.S. Army deployment scheduling patterns. Moreover, a variety of personnel trends, ranging from attrition rates to medical and family issues, have been linked to deployment frequency. While certain levels of persistent operational demand inevitably lead to an increased stress on the force, it is reasonable to conclude that improving deployment measures known to affect personnel is of vital importance.

The U.S. Army frequently acknowledges the importance of stabilization time between deployments and the impact of frequent long deployments. The ratio of the time that a unit is deployed in theater or "boots-on-the-ground" (BOG) to the time not deployed (Dwell) is referred to as a BOG:Dwell ratio. The Secretary of Defense has historically set 1:2 as an objective, whereas the Army prefers a sustained ratio of 1:3. From 2003 to 2011, neither of these goals was met, as the Army deployed soldiers to Iraq and Afghanistan at BOG:Dwell ratios closer to 1:1 than 1:2 (Bonds et al. 2010). During that period, deployments for General Purpose Forces (division and below) supporting named operations outside the continental United States were standardized at 12 months. In an effort to increase the quality of life for Soldiers and families, rebalance the force, and provide better alignment of units, the Army began transitioning to a 9-month deployment period (Department of the Army 2011b).

> The mental health advisory teams sent to Operations Iraqi Freedom and Enduring Freedom have conducted investigations and provided insight and recommendations to improve force health. In November 2006, Mental Health Advisory Team IVs central findings included the following observations: Overall, Soldiers had higher rates of mental health problems than Marines, and Deployment length was related to higher rates of mental health problems and marital problems. Key recommendations included extending the interval between deployments and decreasing deployment

✉ Mark Zais
mark.zais@colorado.edu

Manuel Laguna
laguna@colorado.edu

[1] University of Colorado at Boulder, Boulder, CO, USA

length. The Army has only focused on the first recommendation.
— Lieutenant Colonel Heather Reed, Military Review May–June 2011

Numerous studies focus on measuring the stress on the force caused by deployments, to include physical and mental health of individuals. However, there is little evidence that current unit sourcing and scheduling methods focus on improving these issues. Sourcing requirements complicate unit scheduling because maximizing the level of operational readiness by seeking to rotate units to familiar regions can conflict with deployment timing goals. This paper proposes a deployment scheduling method, which when compared to traditional deployment scheduling techniques simultaneously improves multiple simultaneous objectives ranging from length and frequency of deployments to operational consistency of locations.

In Sect. 2, we present different types of scheduling problems, to include those specific to military deployments. We expand upon modeling objectives and constraints associated with deployment scheduling and unit assignments in Sect. 3 and discuss the relationship of each those objectives as constructs of scheduling management in Sect. 4. In Sect. 5, we construct the DSUA problem as an integer programming problem and then present a more efficient method of graph coloring in Sect. 6 that is easily adaptable to objective priorities. Finally, we use historical demand scenarios to conduct a computational experiment and compare algorithms in Sect. 7.

2 Scheduling problems

The scheduling literature is one of the richest in operations research. However, traditional scheduling problems such as job scheduling, machine scheduling, and employee scheduling problems do not provide a convenient fit to achieve the goals of military deployment scheduling. Moreover, the small amount of specialized literature related to military deployment scheduling is focused on lower level planning issues such as route and logistics scheduling.

Machine scheduling and production scheduling problems vary in complexity. Lenstra et al. (1977) show that while some classical machine scheduling problems are efficiently solvable, others are NP-hard. For instance, Cheng and Chen (1994) consider sequencing a set of simultaneously available jobs on several identical parallel-machines with earliness and tardiness penalties. The objective is to minimize some penalty function of earliness, tardiness, and due-date values. This NP-hard problem is similar but not exactly the same as the DSUA problem where jobs represent units and machines may be viewed as locations.

Modeling of personnel or employee scheduling problems appear to have structural requirements and constraints more similar to the deployment scheduling problem. Personnel scheduling problems have been studied extensively. According to Baker (1976), three main groups can be distinguished: shift scheduling, days off scheduling, and tour scheduling. The latter combines the first two types. Alfares (2004) proposes ten categorical approaches for solving tour scheduling problems, ranging from integer/linear/goal programming to metaheuristics.

Glover and McMillan (1986) present the general employee scheduling problem which extends the standard shift scheduling problem by discarding key limitations and overcoming large scale complexity by generating solutions with a combination of management science and artificial intelligence methods. Glover (1989) also shows that tabu search algorithms are an applicable strategy for combinatorial optimization problems that involve scheduling. Tabu search algorithms have been applied to scheduling problems ranging from distributing workloads among machines to large scale employee scheduling. As mentioned by Glover (1989), tabu search methods possess the capability of being merged with other methods. Like machine scheduling problems, employee scheduling problems do not precisely fit the DSUA problem because of unique constraints and objectives.

Complex combinatorial scheduling problems have been approached with graph coloring methods. As we identify later in this paper, the structural nature of some scheduling problems makes graph coloring an attractive formulation. Gamach et al. (2007) use graph coloring methods to determine a feasible schedule for crew scheduling problem within the airline industry. Moreover, they propose a new methodology to determine the existence of a feasible solution based on a graph coloring model and and a tabu search algorithm. Building upon observations by Gamach et al. (2007), we adapt graph coloring to the unique structure, goals, and constraints of the deployment scheduling described below.

2.1 Deployment scheduling and unit assignment problems

Hodgson et al. (2004) developed a deployment scheduling analysis tool (DSAT) for studying military deployment scenarios and optimizing the scheduling of transportation assets. The DSAT enhances exiting deployment schedules with heuristics for equipment and transportation routing, but does not optimize the top-level unit deployment schedule. McKinzie and Barnes (2004) surveyed the numerous legacy and current strategic mobility models of the time period. While varying in intent, each focuses on some aspect of the defense transportation system and treat unit deployment schedules or time phased force deployment data (TPFDD) as an input. Aviles (1995) created an integer programming model to develop deployment schedules from two nearly simultaneous major regional conflicts using a sequential heuristic. This method meets the initial demands but does not extend to unit deployment cycles during persistent conflict.

2.2 Unit deployment scheduling

Currently, Forces Command (FORSCOM) produces a deployment schedule from the ARFORGEN Synchronization Tool, which informs subsequent sourcing tools. The output from FORSCOMs ARFORGEN Synchronization Tool provides critical arrival and return dates for each rotating unit (Hughes et al. 2011). However, units are assigned at rolling time intervals and do not incorporate analytical methods to optimize manning objectives, which will be described in Sect. 3.1.

The ARFORGEN Synchronization Tool simulates discrete events to achieve a predictive view of the Army inventory moving through the ARFORGEN process over time. It integrates data warehousing, discrete event modeling, scheduling, optimization algorithms, and data visualization into a scenario management infrastructure for sourcing. However, this tool requires a human interaction for selecting sourcing procedures and is most effective for "what—if" analysis or synchronizing processes without necessarily adhering to a strategic optimization goal.

One of the most common measurements for stress on the force is the BOG over dwell ratio. For the active component of the U.S. Army, the BOG:Dwell ratio is the ratio of deployed periods to non-deployed periods. Theoretical goals are often defined in *steady-state* scenarios. A steady-state rotation occurs when the amount of forces in the available force pool exceeds requirements (supply exceeds demand). A steady-state rotation roughly corresponds to the global steady-state security posture and enables the Army to generally achieve operational cycle rotations (Department of the Army 2011a). Most deployment polices are generally tailored to steady-state parameters unless specified otherwise.

The Army Force Generation regulation, (Department of the Army 2011a), also describes a *surge* rotation as a scenario where demand exceeds forces in the available force pool (at steady-state rotational rates). The Army responds through a surge of additional deploying units from an otherwise non-available state (training phase). A surge rotation is characterized by operational cycle rotation ratios shorter than described in the steady-state rotation and by reduced capabilities due to shortened preparation times before deployment.

As mentioned earlier, the Army's steady-state unit BOG:Dwell ratio goal is 1:3. In a 36-month cycle, active Army units will be available for deployment for 9 months (one period) and be in Dwell for 27 months (three periods). The Army's unit BOG:Dwell ratio goal during surge rotations is 1:2. In a 36-month cycle, active Army units will be in the available phase for 12 months (one period) and be in Dwell for 24 months (two periods) (Department of the Army 2011a). Historically, and specifically in the period of this analysis (2003–2011), the Army has failed to consistently achieve these goals.

Competing with the above-mentioned goal of reduced frequency is the desire to reduce deployment lengths for individual units. The Army's standard deployment during the period of this analysis was a 12-month deployment. However, during a period of surge (2007–2008), deployments were increased to 15 months. Beginning on January 1, 2012, Army policy redefined the standard deployment length to 9 months in an effort to reduce the impact of lengthy deployments on individuals and families (Department of the Army 2011b). While the feasibility was dictated by an observed and projected decrease in demand, we demonstrate in Sect. 4 that reduced deployment lengths impact the theoretical lower bounds for achievable unit BOG:Dwell ratios.

3 Scheduling objectives and constraints

Modeling methods for DSUA efficiency extend beyond just improving BOG:Dwell ratio goals. Analytical methods and heuristics can incorporate other desired objectives such as improved location consistency and reduced unit supply criteria. In order to improve deployment scheduling metrics, it is useful to identify the key data, constraints, and objectives of the problem in a similar manner as modeling staff scheduling problems as presented by Blöchliger (2004). The DSUA problem incorporates objectives similar to a general employee scheduling problem, such as minimizing the number of resources (employees versus units); however, it introduces some unique components in terms of variable deployment lengths and location-based operational readiness.

3.1 Deployment scheduling objectives

Army regulations and publications stipulate specific goals and policies regarding deployment scheduling metrics. Most publicly addressed are goals pertaining to length of deployments and BOG:Dwell ratios. However, location consistency is equally important in terms of readiness and enveloping operational and strategic objectives. Moreover, location consistency can be at direct odd with heuristics and algorithms that are intended to improve other scheduling objectives. The following objectives are currently identified as important metrics for evaluating a deployment schedule:

1. Minimize the number of units required to source a deployment schedule,
2. Minimize the number of deployed locations per unit,

3. Minimize the length of deployments.

The first two objectives are directly targeted using optimization methods, while the third objective is manipulated as an input parameter to our approaches. Because the length of deployment varies in a very limited integer range, optimizing on the parameter is rather straight forward. For example, assuming a maximum range of 9–15 months, we can formulate each instance of the parameter for evaluation.

We can measure the merit of a schedule by one or more of these objectives. While we discuss each of these objectives in other sections, we emphasize the importance of the the first stated objective, which minimizes the size of the required sourcing pool. Minimizing the number of units required has significant structural, policy and budget implications. Clearly, demonstrating theoretical capabilities given a set of constraints is useful for comparison with historical data and in guiding future decisions about the size of a force required to meet potential demand scenarios.

Lastly, we are concerned about the ratio of deployment time to dwell time per unit. It is not included above as an optimization objective; however, it is critical that we measure and assess the performance of each approach in regards to BOG:Dwell ratio goals dictated by policy. Minimum BOG:Dwell ratios not only exist as a constraint for the unit assignments during souring, but we also assess each approach related to the average ratio for all units as described earlier for surge and steady-state scenarios.

3.2 Demand constraints

Arguably, the most important constraint in any scheduling problem is the ability to meet demand. In the case of the deployment scheduling problem, demand is defined by mission requirements that are often difficult to predict. Previously, we discussed how deployment goals can change under steady-state versus surge demand scenarios. While steady-state scenarios are commonly used for strategic planning and theoretical limits, they are rarely observed in practice.

Figure 1 shows the actual deployment strength of battalion-sized combat units during 86 months of Operation Iraqi Freedom. For analysis purposes, we use the historical deployment pattern as a proxy for demand and assume that an excess in unit supply was not deployed. This demand figure is limited to infantry, armor, cavalry, field artillery, and engineer units. The colors depict the separation in demand by province location. It is clear that during this period of persistent conflict, steady-state demand did not exist.

3.3 Deployment constraints

The Army established a dwell versus deployment policy to compliment the previously mentioned ratio goals. Prior to the 2012 calendar year, individuals deployed for a period of 12 months or more were required to subsequently dwell for at least 1 month per month deployed (Department of the Army 2011b). In other words, the upper bound BOG:Dwell ratio is 1:1. For the purposes of developing methods to compare with historical data, a modeling assumption is made that units must dwell for a duration of time greater than or equal to the previous length of deployment.

4 Scheduling management

Prior to the development and evaluation of methods to optimize the objectives in Sect. 3.1, it is useful to understand the limits which can be derived based on the relationship of the first two objectives when presented as variables with demand and units of supply. Section 4.1 presents a closed-form solution of the best average BOG:Dwell ratio for units given a set of fixed parameters in steady-state rotation. Section 4.2 presents a lemma to identify the unique combination of group sizes that maximize BOG:Dwell when units are rotated within groups rather than mixed independently.

4.1 Closed-form constraints

A unit's BOG time is calculated at the time it arrives in a deployment theater until the time it departs. Mentioned previously, Army leadership frequently references a goal of maintaining a BOG:Dwell ratio of less than $1:2$ during other than steady-state operations.

The BOG:Dwell ratio is a supply and demand issue. It is easy to prove that when demand is constant the ratio converges to a theoretical limit (Dabkowski et al. 2009). The following notation is used for the derivation of BOG:Dwell ratios and examining the limits:

N: number of rotational units
M: number of sustained units required to support deployment theater
x: duration of deployment ($x \in R+$)
n: duration of the overlap between a deployed unit and its replacement unit (same time units as x; $n \in R+, n < x$)
T: length of the analysis period (same time units as x; $T \in R+, T > x - n$)

In a previous report in which we analyze the unit and individual BOG:Dwell rates during periods of steady-state demand, (Dabkowski et al. 2009) construct the following theorem and proof to show the theoretical limits:

Theorem 1 *As T tends to infinity, if* $\frac{N}{M} > 1 + \frac{n}{x-n}$, *the unit BOG:Dwell ratio is given by the relation*:

$$r(T) = 1 : \frac{N(x-n)}{Mx} - 1$$

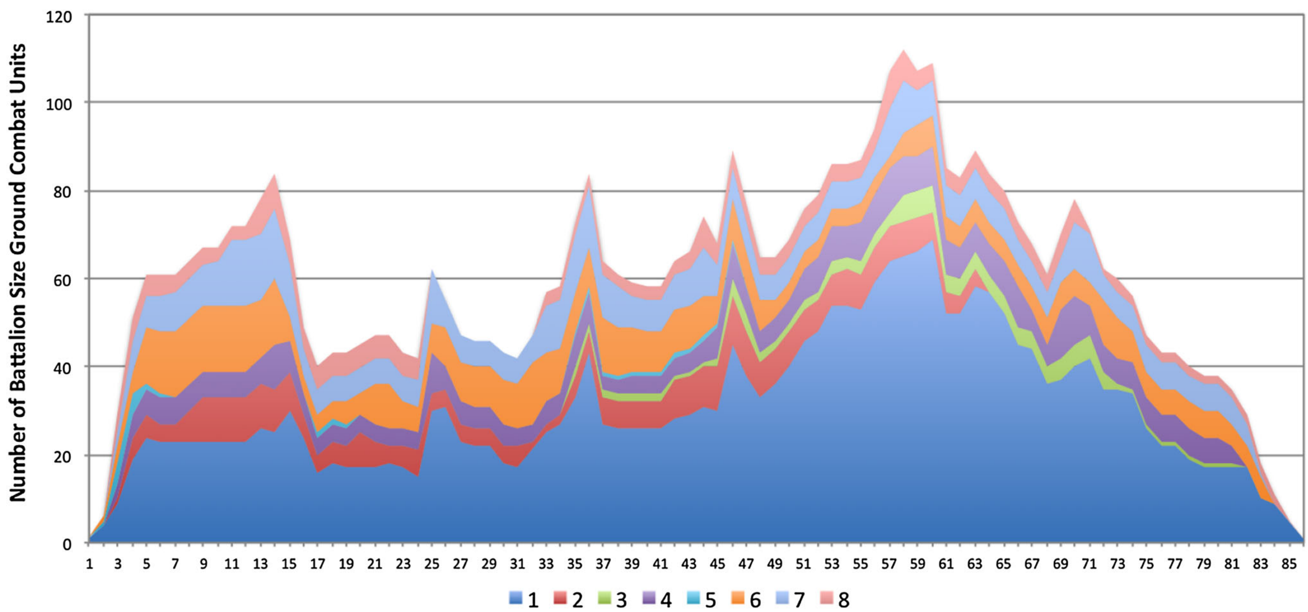

Fig. 1 Monthly unit demand (colored by location)

We can demonstrate an application of the limiting theorem presented above. Consider a scenario in which we have 44 units available in supply for rotation and a mandatory overlap of 40 days among rotating units. Figure 2 presents the results of the theorem when parameters of demand and length of deployment are adjusted. For example, as highlighted, the only way that a ratio of 1:2 or better can theoretically be achieved with 12 month deployments is if demand is less than or equal to 13 units in steady-state. Highlighted in red, the figure shows the best theoretical right-hand side (RHS) of a BOG:Dwell ratio (1:RHS) given the existing deployment-length policy and "Available" demand planning requirement in 2009.

4.2 Unit grouping lemma

Another way to address deployment rotations is to deploy units as groups. More simply, a unit would belong to a group of units that deploys in a rotational cycle such that the units ensure consistency in the same locations. Another closed-form best-case solution for rotational frequency can be obtained if units are grouped together in this manner. This rotational pattern can not achieve a better theoretical limit than Theorem 1, but it does allow us to increase operational predictability and address location consistency. The following Lemma is constructed and proven in the document by Dabkowski et al. (2009):

Lemma 1 *Unit-grouping Lemma. Given N rotational units and a sustained requirement of M units such that $M > 0$, there exists a unique combination of $(M - s)s$ and $r(s + 1)$ groups of units that fulfill the requirement, where $r = N - M\lfloor N/M \rfloor$ and $s = \lfloor N/M \rfloor$.*

4.3 Location consistency

During persistent conflict, maintaining a certain level of consistency for units deploying to a location has obvious tactical and operational advantages. Familiarity with a location provides a benefit to operational readiness. As stated by Reed (2011), a unit returning to the same location (with sufficient continuity in the organization) has a more positive effect in maintaining continuity of operations than a new unit or one with high turnover. Moreover, if units return to a location where they have previously served and they retain some of their leaders, they have the knowledge necessary to anticipate the second-order and third-order effects of their actions and are less likely to derail efforts of a previous unit.

The risks associated with unit transitions and handovers are a documented concern in counterinsurgency operations (Department of the Army 2006). Moreover, The US has recognized that enduring relationships and and an understanding of the environment are critical aspects of recent conflicts. Continuity is ensured with effective transitions and with an instilled level of familiarity that can be increased with targeted unit rotations. Often, developed knowledge is used inefficiently while at the same time we task individuals with preparing for deployment by studying the topography, people, and culture of a region (Kilcullen 2006).

It is important to include location consistency with the DSUA problem because a goal of location consistency can be at odds with other objectives. Section 4.2 shows that main-

Available BCT Requirement

Length of Deployment (Months) BOG	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
6	3.30	2.82	2.44	2.13	1.87	1.64	1.46	1.29	1.15	1.02	0.91	0.81	0.72	0.64	0.56	0.49	0.43
7	3.47	2.97	2.58	2.25	1.98	1.75	1.55	1.38	1.23	1.10	0.99	0.88	0.79	0.70	0.63	0.55	0.49
8	3.60	3.09	2.68	2.34	2.07	1.83	1.63	1.45	1.30	1.16	1.04	0.94	0.84	0.75	0.67	0.60	0.53
9	3.70	3.18	2.76	2.42	2.13	1.89	1.68	1.51	1.35	1.21	1.09	0.98	0.88	0.79	0.71	0.63	0.57
10	3.78	3.25	2.82	2.48	2.19	1.94	1.73	1.55	1.39	1.25	1.12	1.01	0.91	0.82	0.74	0.66	0.59
11	3.84	3.31	2.88	2.52	2.23	1.98	1.77	1.58	1.42	1.28	1.15	1.04	0.94	0.85	0.76	0.68	0.61
12	3.90	3.35	2.92	2.56	2.27	**2.01**	1.80	1.61	1.45	1.31	1.18	1.06	0.96	0.87	0.78	0.70	0.63
13	3.95	3.40	2.96	2.60	2.30	2.04	1.83	1.64	1.47	1.33	1.20	1.08	0.98	0.88	0.80	0.72	0.65
14	3.98	3.43	2.99	2.63	2.32	2.07	1.85	1.66	1.49	1.35	1.22	1.10	0.99	0.90	0.81	0.73	0.66
15	4.02	3.46	3.02	2.65	2.35	2.09	1.87	1.68	1.51	1.36	1.23	1.11	1.01	0.91	0.83	0.75	0.67
16	4.05	3.49	3.04	2.67	2.37	2.11	1.89	1.69	1.52	1.38	1.24	1.13	1.02	0.92	0.84	0.76	0.68
17	4.08	3.51	3.06	2.69	2.38	2.12	1.90	1.71	1.54	1.39	1.26	1.14	1.03	0.93	0.85	0.77	0.69
18	4.10	3.53	3.08	2.71	2.40	2.14	1.91	1.72	1.55	1.40	1.27	1.15	1.04	0.94	0.85	0.77	0.70
19	4.12	3.55	3.10	2.72	2.41	2.15	1.93	1.73	1.56	1.41	1.28	1.16	1.05	0.95	0.86	0.78	0.71
20	4.14	3.57	3.11	2.74	2.43	2.16	1.94	1.74	1.57	1.42	1.28	1.16	1.06	0.96	0.87	0.79	0.71
21	4.16	3.58	3.13	2.75	2.44	2.17	1.95	1.75	1.58	1.43	1.29	1.17	1.06	0.96	0.88	0.79	0.72
22	4.17	3.60	3.14	2.76	2.45	2.18	1.96	1.76	1.59	1.43	1.30	1.18	1.07	0.97	0.88	0.80	0.72
23	4.19	3.61	3.15	2.77	2.46	2.19	1.96	1.77	1.59	1.44	1.31	1.18	1.07	0.98	0.89	0.80	0.73
24	4.20	3.62	3.16	2.78	2.47	2.20	1.97	1.77	1.60	1.45	1.31	1.19	1.08	0.98	0.89	0.81	0.73

1 : Dwell Ratio; Assuming 45 BCTs (44 Rotational) and 40 day RIP/TOA Length

Fig. 2 BOG:Dwell ratios with fixed unit supply and overlap

taining unit grouping reduces the theoretical limitations of BOG:Dwell ratios. Location consistency also has the effect of increasing the lower bound of an objective function that minimizes the number of units used for scheduling.

Deployment scheduling logistics articles reference parameters for general theater locations during simultaneous conflicts but in general do not address location scheduling parameters within a theater of operations. Location consistency is generally regarded as a readiness issue; its impact to scheduling logistics is primarily isolated to the high-level unit scheduling problem addressed here.

5 Scheduling optimization

The DSUA problem can be set up as an integer programming problem similar to those used in employee scheduling (Glover and McMillan 1986) and staff scheduling problems (Blöchliger 2004). While it is useful to demonstrate the ability to present the deployment scheduling problem in terms of integer programming properties, later we show that specific characteristics of this problem lend it to be solved more efficiently with graph coloring approaches. Initially, we define the following scheduling parameters for the binary integer program:

- q: unit identifier,
- L: number of deployment locations (by province),
- w: minimum dwell time policy ($w \in \mathbb{Z}+$),
- d: length of deployment in months ($d \in \mathbb{Z}+$),
- T: length of deployment horizon ($T \in \mathbb{Z}+, T > d$),
- r_{lm}: required number of units in location l in month m ($r \in \mathbb{Z}+$).

The binary variables of this formulation are as follows:

$$x_{qlm} = \begin{cases} 1: \text{if unit } q \text{ is assigned to location } l \\ \quad \text{in month } m, \\ 0: \text{otherwise,} \end{cases}$$

$$y_{qlm} = \begin{cases} 1: \text{if unit } q \text{ starts deployment in location } l \\ \quad \text{in month } m, \\ 0: \text{otherwise,} \end{cases}$$

$$v_{ql} = \begin{cases} 1: \text{if unit } q \text{ is deployed to location } l \\ \quad \text{at any time,} \\ 0: \text{otherwise,} \end{cases}$$

$$z_q = \begin{cases} 1: \text{if unit } q \text{ is ever scheduled for deployment,} \\ 0: \text{otherwise.} \end{cases}$$

The variable x_{qlm} is used to track the utilization of a particular unit at a given time and location. The variable y_{qlm} identifies the starting point of each deployment by time and location. We use v_{ql} to track the number of deployments by unit and location and incorporate z_q to identify if a unit is ever scheduled for deployment.

5.1 Optimization objectives

The deployment scheduling problem can be optimized on multiple objectives, particularly with regard to the number of units required to source the schedule and the number of deployed locations per unit. As mentioned previously with scheduling objective, we treat deployment length, d, as an input parameter rather than an objective due to its limited integer range.[1]

Suppose we have a sourcing pool and want to minimize the number of units used to source the the demand. An optimization objective can be applied to the number of units used in the schedule, which minimizes $\sum_q z_q$. While this objective function can be deemed inequitable for an existing sourcing pool, a unit utilization objective allows us to determine a lower bound for structure requirements, given an expected demand.

Two of the metrics that are useful in measuring readiness and stress on the force are location consistency and BOG:Dwell ratios. The number of locations per unit can be minimized by optimizing on the objective function, $\sum_q \sum_l v_{ql}$, which intuitively works against an objective to minimize units. Creating an objective function that minimizes BOG:Dwell ratios causes complications for an integer programming approach. Assume we can substitute the average deployed time per unit for the average BOG:Dwell ratio since we are optimizing in a finite time horizon. Given N total units, an objective function minimizing the average time deployed

$$\frac{\sum_q \left(\sum_l \sum_m y_{qlm} d \right)}{N}$$

poses nonlinear issues that make it difficult to solve.

5.2 Constraints

The DSUA problem seeks to optimize single or multiple objectives while meeting mission demands at multiple locations over a finite time horizon. For the purposes of this research, time periods are measured in months. The scheduling optimization must meet minimum demand requirements while adhering to deployments policies, specifically pertaining to the length of deployments and minimum dwell time for units. The following constraints are created to depict deployment demand and unit assignment policies:

$$\sum_q x_{qlm} \geq r_{lm} \qquad \forall \;\; l, m, \tag{1}$$

$$\sum_{t=\max\{1, m-d+1\}}^{m} y_{qlt} = x_{qlm} \qquad \forall\, q, l, m, \tag{2}$$

$$\sum_{l=1}^{L} \sum_{t=m}^{m+2d} y_{qlt} \leq 1 \qquad \forall\, q, m, \tag{3}$$

$$z_q \geq x_{qlm} \qquad \forall\, q, l, m, \tag{4}$$

$$\sum_m y_{qlm} \leq T v_{ql} \qquad \forall\, q, l, m, \tag{5}$$

$$x_{qlm}, y_{qlm}, z_q \in \{0, 1\} \qquad \forall\, q, l, m.$$

Constraint (1) ensures that demand is met during each time period at each location. Constraints (2) and (3), respectively, align monthly utilization after deployment start dates and establish minimum dwell times between deployment start dates for a unit. Constraint (4) essentially turns on the binary z_q variable when a unit deploys in any time period or location. Lastly, constraint (5) activates v_{ql} if unit q is deployed to location l at least once.

The scheduling problem formulated as an integer programming problem is difficult to solve due to the size and scale. We use an historical data set from Iraq as a basis for the variable indices. Excluding locations with less than five deployments, there are 8 locations and 86 months in the deployment scenario being modeled. If demand is consistent with the historical use of over 200 different units, the problem requires more than 275,000 binary x_{qlm} and y_{qlm} variables combined. Moreover, the number of row constraints associated with (2) and (4) exceed 275,000 and constraints (1) and (5) have 688 and 1,600 rows, respectively. Overall, there are more than 275,000 variables and 300,000 constraints.

The optimization problem is complicated with variable deployment lengths. However, Army deployment policies generally dictate fixed deployment lengths. Deployment length may change at different points in the time horizon but remain consistent among units. Therefore, it is reasonable to assume that deployment lengths cannot vary for each deployment during this analysis. This assumption is important because it lends the deployment scheduling problem to other solution methods, as shown below.

6 Schedule coloring problem

Graph coloring is a method that has useful application in many types of complex problems that involve optimization. The basic objective of graph coloring problems is to assign colors to a set of vertices in a graph such that all vertices are colored with a minimum number of different colors. A constraint that limits two vertices from being colored with the same color is depicted by a connecting edge between the two vertices.

6.1 Definitions and notations

Given a graph $G = (V, E)$ with vertex set V and edge set E, and given an integer k, a k-coloring of G is a function

[1] Unit deployments in persistent conflict range from 9 to 15 months in recent history. It is reasonable to assume a 9-month lower bound since shorter deployments continue to worsen BOG:Dwell ratios and can cause logistics and operations issues that make solutions infeasible in other domains.

Table 1 Example problem: demand by location and month

Location	Month									
	1	2	3	4	5	6	7	8	9	10
1	1	2	2	0	0	0	1	1	1	1
2	0	1	1	1	1	0	0	0	1	1
3	0	0	0	1	1	2	2	2	0	0

$c : V \to \{1, \ldots, k\}$. The value $c(v)$ of a vertex v is called the color of v. The goal of the standard graph coloring problem is to find an assignment of colors to the vertices of G that minimizes the number of colors used such that no two adjacent vertices receive the same color. Colors are defined as positive integers and the number of colors in an optimal coloring solution is called the chromatic number of the graph, denoted by $\chi(G)$. Finding $\chi(G)$ in general graphs is known to be one of the most difficult optimization problems on graphs (Palubeckis 2008).

6.2 Graph coloring for the DSUA problem

When deployment lengths are fixed, the DSUA problem lends itself to be solved more efficiently as a graph coloring problem. As discussed earlier, the Army has policies which standardize the length of deployments. While it is possible for two deployed units to have different deployment lengths, it is a reasonable assumption that deployment lengths are the same for all units during a specified time window. Because this assumption, we can use graph coloring techniques to assign a minimal number of units to deployment schedules. Moreover, we will demonstrate how graph coloring methods can be modified to improve specific objective measures that are unique to the DSUA problem.

Consider a very small-scale example of a deployment schedule in which there is a demand for units in three different locations, $l := 1, 2, 3$ over a time horizon of 10 months. Suppose the demand for each location and time period, r_{lm}, is given as the values in Table 1. We then create a simple algorithm which builds a feasible deployment schedule by assigning deployments sequentially by location to meet each demand 1 month at a time. Hence, if viewed in a matrix format, the schedule is constructed and more importantly labeled by row then column.

Using the demand from Table 1, a simple deployment schedule is resented in Fig. 4 given a standard deployment length of two time periods. The importance of numbering the deployments in a left to right time sequence proves to be critical to the performance of particular graph coloring algorithms. In terms of graph coloring notation, the numbered deployments are the same as graph vertices.

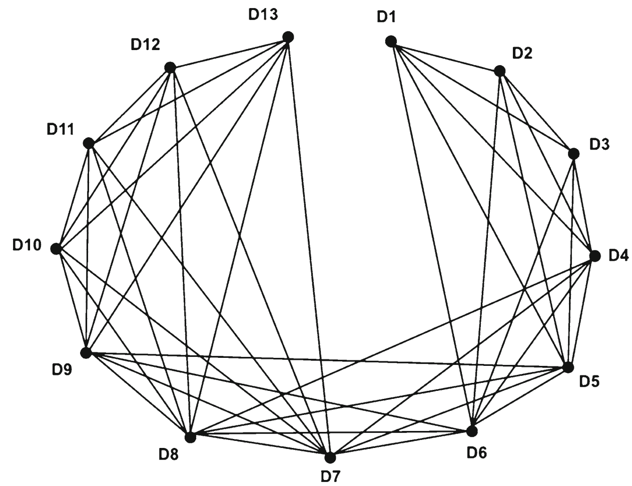

Fig. 3 Graph representation of a deployment schedule

Once again, we refer to the example problem where there are 13 vertices that must be colored. We apply a simple constraint in which two vertices cannot be colored the same unless they are separated by at least two time periods. In deployment terms, this constraint enforces a policy that, if a unit deploys, it cannot be assigned another deployment until is has dwelled for the same amount of time as the previous deployment length. An edge between two vertices represents deployments that are separated by less than two time periods. Figure 3 shows the graph corresponding to the deployments in Fig. 4. The remainder of this section uses this example problem as a basis for describing graph coloring algorithms that are most applicable to the DSUA problem.

6.3 Coloring algorithms for scheduling

Recursive largest first (RLF) is one of the best known coloring algorithms. Developed by Leighton (1979), RLF has been shown to be suitable for large-scale practical problems. First-fit can be considered a special version of RLF that produces optimal solutions for the problem of minimizing the number of colors in an interval graph. We first review how first-fit operates and then describe how we apply it to the DSUA problem.

6.3.1 First-fit algorithm

Let U_1 be the set of uncolored vertices that are not adjacent to any of the vertices colored with color c and U_2 be the set of uncolored vertices adjacent to a least one of the vertices colored with c. Given G from above, the first-fit algorithm chooses the first vertex, v, in U_1 and colors it with the current color c. Upon updating U_1 and U_2, we again choose the first v in U_1. The process continues until U_1 is empty and then is repeated for new colors until all vertices in G are colored.

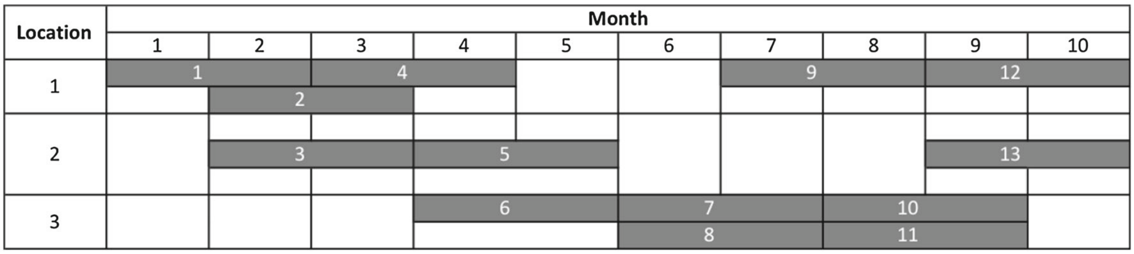

Fig. 4 Feasible deployment schedule without assignment

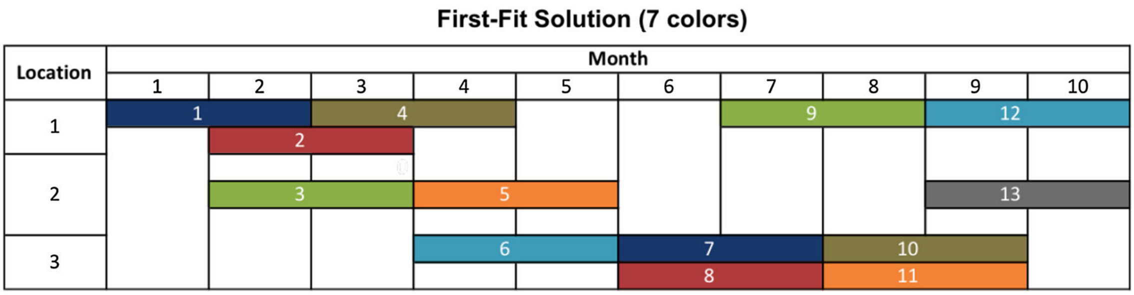

Fig. 5 Minimize the number of units (colors)

The first-fit algorithm produces heuristic solutions for general graphs. However, with a specific structuring of the vertices being colored, the first-fit algorithm produces optimal solutions to the problem of minimizing the number of colors. Specifically, this occurs in problems possessing the same properties of a interval graph. If C is defined as the set of vertices that have already been colored and V' is a subset of uncolored vertices in V, the first-fit algorithm can be depicted as in Algorithm 1.

Set $C = \emptyset,\ U_1 := V,\ U_2 = \emptyset,\ c := 0$;
while $|C| < |V|$ **do**

  if $U_1 \neq \emptyset$ **then**

    Identify first v (lowest integer) in the subgraph of U_1;
    Assign color c to v;
    Move v from U_1 to C;
    Move all $v' \in U_1$ that are adjacent to v from U_1 to U_2;
    $V' = \{U_1, U_2\}$;
  else
    Increment c by 1;
    Set $U_1 := U_2, U_2 := \emptyset$;
  end
end

Algorithm 1: First-fit algorithm

Kierstead (1988) proposed that a graph G is an interval graph if its vertex set can be represented by a collection of closed intervals V of the real numbers so that for all $i, j \in V$, $i \cap j \neq \emptyset$ if and only if the vertex represented by i is adjacent to the vertex represented by j such that the V intervals are a representation of G.[2] Kierstead describes interval graphs as perfect in that the chromatic number $\chi(G)$ of an interval graph G is equal to its maximum clique size $\omega(G)$.

Because of the known interval graph structure of the DSUA problem, the first-fit algorithm is chosen as the basis for solving the variants of the problem that we consider. Further validation of the DSUA problem with the first-fit algorithm in terms of interval graph theory is described in Sect. 6.4.1. Figure 5 shows the coloring solution for the schedule in Fig. 4 using the first-fit algorithm to minimize the number of colors used. First-fit provides an optimal solution of seven colors; however, it is important to note that it is not the only optimal solution.

6.3.2 Color swapping

As mentioned above, first-fit optimizes objective 1, that is, if minimizing the number of units required to source a deployment schedule. However, it does not produce optimal solutions when considering the minimization of deployed locations per unit. Therefore, we search for solutions that minimize our second objective by swapping colors in colored graphs. So, color swapping is applied as a post-processing local search to a colored graph. Swaps are made within a solution of fixed $|C|$ colors and a new solution is evaluated based on a metric other than the number of colors used. In the case of the deployment scheduling problem, we use a local search to improve upon the metric of number of deployed locations per unit (color).

Consider a solution produced by the first-fit algorithm. Each deployment is represented by a vertex, which is assigned a color. A location index is also associated with each

[2] Intervals in the graph may have the same endpoints.

vertex. The local search iterates with each vertex, v, first identifying the color, $c(v)$ and location $l(v)$ indices. With each iteration, a search is conducted to identify another vertex, v' with the same color assignment that is in a different location; therefore, $c(v) = c(v')$ and $l(v) \neq l(v')$. Another neighborhood search is conducted to find a vertex, v'', that is assigned a different color but in the same location as v; therefore, $c(v) \neq c(v'')$ and $l(v) = l(v'')$. If v' and v'' are not connected by an edge and if v and v'' are also not connected by an edge, the colors assigned to v' and v'' are swapped. Using our notation:

$$\text{if } (v', v'') \notin E \text{ and } (v, v'') \notin E$$
$$\text{then } c(v') \leftrightarrow c(v'').$$

Note that after the swap, v and v'' have the same color. Since these deployments occur in the lame location, the swap has assigned the unit to be deployed to a single location instead of two different locations.

The process is repeated for every vertex v until no swaps are possible. The new solution is then subject to the swapping heuristic beginning at the first deployment vertex. If the heuristic passes through a full iteration of all vertices without making any swaps , the heuristic is terminated and the current solution is returned as the best solution after swapping. Consider a solution in which C represents an initial assignment and n is the number of deployments, then the color swapping algorithm is shown in Algorithm 2.

```
Given C;
while Terminate = False do
    Terminate ← True;
    for v = 1, ..., n do
        for v' ≠ v do
            if c(v) ≠ c(v') and l(v) ≠ l(v') then
                for v'' ≠ v and v'' ≠ v' do
                    if c(v) ≠ c(v'') and l(v) ≠ l(v'') then
                        c(v') ↔ c(v'');
                        Terminate ← False;
                    end
                end
            end
        end
    end
end
```

Algorithm 2: Color swapping heuristic

6.3.3 First-fit by location (FFL) algorithm

In the previous section, the first-fit algorithm is followed by a color swapping procedure. Sequencing the first-fit and color swapping steps is a form of lexicographic multi-objective optimization. Specifically, with the lexicographic method, the objective functions are sequenced in order of importance and the optimization problems are solved one at a time (Marler and Arora 2004).

The first-fit by location (FFL) algorithm is a modification of the first-fit algorithm and seeks to minimize both objectives at the same time: the number of colors used to color the vertices of G, and the number of locations assigned to each unit. In each color iteration, the first uncolored vertex, v is colored with the next color, c. Subsequently, rather than choosing the first vertex in the U_1 set, we choose the first vertex in U_1 that has the same location as v and color it with c. In the case where U_1 is not empty but does not contain a vertex with the same location as v, the first vertex in U_1 is colored c. The U_1 and U_2 sets are updated after each coloring and the process is repeated until U_1 is empty and the next iteration begins with the next color, (i.e., $c = c + 1$).

Figure 6 demonstrates the different coloring solutions for the uncolored schedule in Fig. 4 using first-fit with swapping and FFL algorithms. Despite the small scale of this example (13 vertices), the two algorithms clearly produce different solutions, as first-fit with swapping addresses the objectives lexicographically, while the FFL seeks to improve objectives simultaneously. FFL can also be combined with the swapping heuristic to seek an improved solution on the location objective. A FFL swapping solution is not presented in Fig. 6 since the scale is too small for swapping to produce an improvement.

6.4 Algorithm comparisons

In this section, we discuss the theoretical characteristics of the above-mentioned algorithms and demonstrate their performance with the initial toy problem before comparing results in the computational experiments in Sect. 7. We begin with the observation that the DSUA problem falls into a subset of interval graph problems. Accordingly, the first-fit algorithm provides an optimal solution for the unit minimization problem. Extending the theoretical implications of first-fit, it is clear that the FFL algorithm produces a solution which is always equal or better than the first-fit with swapping solution when measured against the additional location objective. However, FFL demonstrates an ability to meet or exceed the performance of first-fit when measuring the performance of average BOG:Dwell ratios prior to any swapping heuristics.

6.4.1 Interval graph theory

The deployment schedule, as represented earlier in Fig. 4, has the characteristics of an interval graph. Using time periods as an index, the vertices in the graph are structured with a left end point ordering. Moreover, the graph can be described as

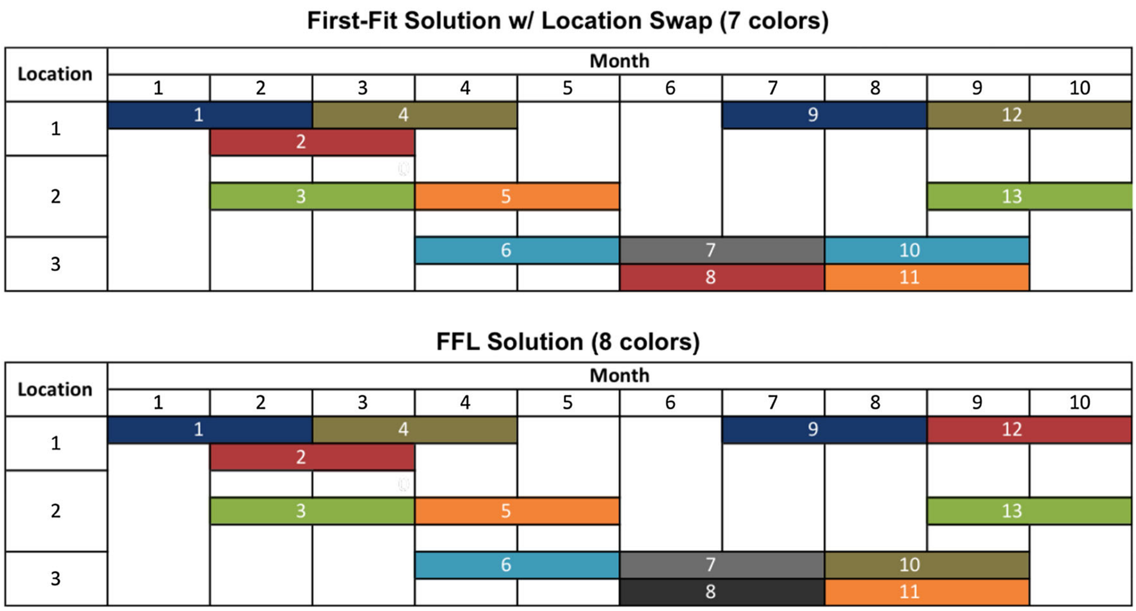

Fig. 6 Minimize the number of units (colors) and locations

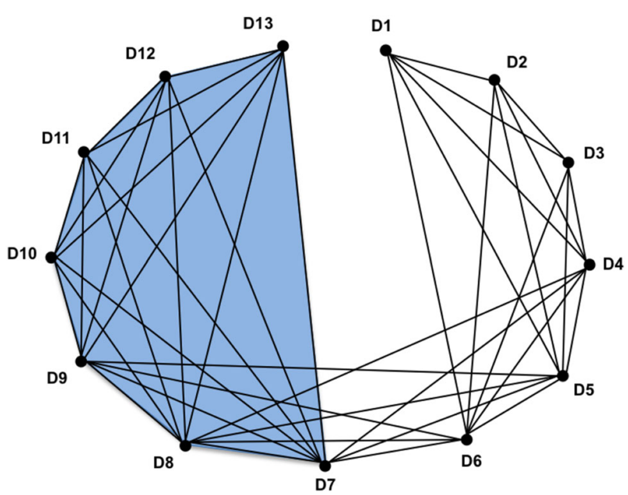

Fig. 7 Optimal coloring of G equals maximum clique size ($\omega(G) = 7$)

a collection of closed intervals ordered in a relation such that $i < j$ iff the right end point of i is less than the left end point of j (Kierstead and Qin 1995). As such, the first-fit provides an optimal solution equal to the maximum clique size.

Returning to the example problem, the maximum clique size of the graph is simple to calculate. Graphically it is represented in Fig. 7 as the set of vertices that are shaded in blue. The maximum clique size, $\omega(G) = 7$, is equal to the optimal solution when minimizing colors with first-fit.

6.4.2 Performance on objectives

This section shows how the different algorithms perform against the measured objectives, prior to analysis with large scale and historical data sets in the following section. Using the example problem we replace the terms vertices and colors for deployments (deploy) and units respectively. Although the swapping heuristic is an extension to the first-fit algorithm, first-fit and first-fit swapping are compared as separate algorithms since the first has a single objective and the latter deals with two objectives lexicographically. FFL is the third comparison.

Naturally, the performance of these algorithms is evaluated like all graph coloring problems in terms of minimizing the number of units (colors) used to meet deployment (vertex coloring) requirements. Performance is compared by the average number of locations assigned per unit during the deployment horizon. It is also worth noting the maximum number of locations for any unit, which could indicate a skewed distribution among units. The algorithms are also evaluated based on BOG:Dwell ratio statics, most importantly the average unit ratio. BOG:Dwell ratios are only measured when a unit has more than one deployment. The BOG:Dwell statistic, as shown in Table 2, is the RHS of the BOG:Dwell ratio with the left-hand side equal to 1; therefore, in the format of 1:Dwell.

It is clear from Table 2 that even with a small example, FFL does not guarantee the same minimum unit optimality as first-fit. First-fit swapping and FFL significantly improve the locations per unit statistic, but FFL intuitively dominates due to the availability of additional unit resources. The average dwell time is also improved by both algorithms as residual benefit from the location improvement. Focussing on FFL, it never produces a worse BOG:Dwell statistic than first-fit. This is obvious because first-fit colors vertices based on the parameter of interval size, which is essentially a lower bound.

Table 2 Minimizing the number of units and locations

Method	Deploy	Edges	Units	Loc/unit	Diff (%)	Avg Dwell	Diff (%)
First-fit	13	44	7	1.8571	–	1.3333	–
First-fit swap	13	44	7	1.2857	−30.77	1.5833	+18.75
FFL	13	44	8	1.2500	−32.69	1.9000	+42.50

Table 3 Comparison of algorithms to historical data

Method	Deploy	Units	Time (sec)	Loc/unit	Avg Dwell	Max Dwell
$w = 1{:}1$						
First-fit	435	192	17	1.7656	1.3770	1.7500
First-fit swap	–	–	17	1.2343	1.4893	2.9167
FFL	435	196	19	1.2245	1.4843	2.4167
FFL swap	–	–	17	1.2194	1.5485	3.1667
Actual	460	203	–	1.8252	1.4166	3.0909
$w = 1{:}2$						
First-fit	435	249	27	1.5301	2.3109	2.6667
First-fit swap	–	–	6	1.1606	2.3888	4.1667
FFL	435	258	35	1.1395	2.4983	4.5000
FFL swap	–	–	3	1.0969	2.4826	4.4167

FFL can extend into the next interval to find a better location, thereby increasing dwell time. Generally, FFL performs better than first-fit swapping in average dwell time but is not guaranteed a better result. For the example shown in Table 2, the maximum dwell time is 1.5 using first-fit and 2.5 using both location improving heuristics.

7 Computational experiment

The computational experiment applies the abovementioned algorithms to a historical data set that represents an 86-month demand scenario in Iraq from 2003 to 2010. Historical deployments are used as a proxy for demand in 8 separate provinces which separate demand by location. The historical demand, depicted earlier in Fig. 1, represents a selection of unit types and is not a steady-state scenario. In this section we compare the historical deployment schedule, which used 203 units, with schedule solutions produced by each of the graph coloring algorithms presented earlier.

In addition to comparing results with the historical scenario, we evaluate the performance and sensitivity of solutions when we adjust parameters such as deployment lengths and minimum dwell policies. Subsequently we present an analysis of the algorithms with different demand scenarios, to include steady-state demand. The analysis is concluded with a validation of first-fit as an optimal procedure for the objective of minimizing units.

All algorithms and computations were programmed using Java code in Netbeans IDE and executed for comparison on a computer with a 2.7 GHz Intel Core i7 processor. Each algorithm is automated to accept different sizes of demand input and adjusts to user-defined parameters.

7.1 Historical comparison

The historical deployment schedule actually assigned 460 individual deployments to satisfy the demand requirements. The deployments were fulfilled by 203 different operational units, which were all battalion size elements. During the time horizon of this analysis the standard length of deployments was 12 months; however, during a surge period, deployments lengths extended to 15 months. Assuming that the demand is known and deployments are fixed at 12/15 months depending on the policy at the time, we are able to construct a deployment scenario that meets the same demand using 435 deployment assignments. Using a maximum BOG:Dwell policy of 1:1, the schedule's corresponding graph has 52,359 edges.

The results in Table 3 indicate that demand could be met with a minimum of 192 units by adhering to a 1:1 BOG:Dwell ratio policy. Moreover, the number of locations per unit could be minimized using a sourcing plan of 196 units and the FFL heuristic with swapping. We also observe that when the swapping heuristic is applied to first-fit, the average number of locations per unit is less than 0.02 greater than FFL swap. The swapping heuristic appears to introduce some inconsistencies in the comparison of first-fit swap and FFL swap as measured against average dwell, but the differences are relatively small.

In the lower half of the table, the minimum dwell policy is increased to twice the length of deployment. The number

Table 4 Number of units: sensitivity to deployment lengths

		Deployment Length (d = months)						
Method	w	9	10	11	12	13	14	15
First-fit	1:1	206	210	204	207	209	213	192
FFL		209	218	213	212	214	216	192
First-fit	1:2	292	297	283	287	283	274	249
FFL		296	305	306	295	285	284	267
First-fit	1:3	369	375	355	351	343	338	341
FFL		387	393	378	356	343	339	341
Number of deployments		693	642	590	507	468	436	406

of edges increase in the graph to 69,067 and the number of units required increases accordingly. Most importantly, in order to adhere to BOG:Dwell constraint of 1:2 for all units, the number of required units increases to 249 or 258 for first-fit and FFL solutions respectively. Another less intuitive dynamic, which remains consistent in subsequent scenarios, relates to the locations per unit statistic. As the minimum dwell policy increases, the number of locations per unit can be further reduced.

7.2 Deployment length and minimum Dwell

This section examines the sensitivity of the coloring solutions to changes in the deployment length parameter and minimum dwell constraint. Unlike the solutions in Table 3, the deployment length is constant throughout the time horizon. Table 4 shows the number of units used to color the deployment schedules using the first-fit and FFL algorithms. As the minimum RHS dwell constraint increases from 1 to 3, the number of units increases. However, the direction of change varies with increases to deployment lengths.

In Table 5, the location objective is also measured with changes to the deployment length parameter and dwell constraint. Results are shown post-swapping for first-fit and FFL and compared to the base first-fit results which only target the number of units objective. Both algorithms with post-swapping heuristics significantly outperform standard first-fit. Moreover, FFL swap almost always produces a marginally better solution than first-fit swap and never does worse.

The unit and location objectives are clearly in conflict with each other. The average dwell statistic is not specifically targeted by either of the coloring algorithms or the color swapping heuristic. However, we are able to measure the performance of our solutions in terms of the average dwell (RHS of BOG:Dwell) statistic. Arguably, BOG:Dwell statistics are weighed with equal importance as unit and location statistics. Therefore, it is significant that our FFL and swapping solutions generally improve dwell statistics beyond first-fit as a by-product of targeting other objectives.

Table 6 shows the results of first-fit and FFL as well as post-swapping results from both. The result which provides the best solution for each combination of deployment length and minimum dwell time is highlighted in bold. Results from FFL before and after swapping are displayed since the swapping heuristic does not guarantee any improvement to average dwell. In fact, as the minimum dwell policy constraint is increased from 1:1 to 1:2, the swapping heuristic appears to shift from increasing to decreasing the average dwell time. The three latter solutions dominate first-fit in all parameter and constraint scenarios until the maximum deployment length and dwell policy limit is reached.

The dwell statistic is significant because, as mentioned in Sect. 3, success is measured based on the average BOG:Dwell ratios of units rather than an enforced minimum dwell policy. For example, the minimum dwell policy could be a ratio of 1:1 yet the Army may have a goal of maintaining an average BOG:Dwell ratio of 1:2.

Given 12-month deployment lengths, Fig. 8 demonstrates the relationship of our coloring methods, minimum dwell constraints and published BOG:Dwell ratio goals. As the minimum dwell constraint is raised as depicted by the horizontal axis, the top half of the graph shows the average dwell statistics after first-fit and first-fit swap. The lower half of the figure shows the same trend for FFL and FFL swap. The vertical grid lines identify the points at which solutions achieve goals of 1:2 and 1:3 average BOG:Dwell ratios.

The density of deployments depicted in Fig. 8 is consistently 507, while the number of edges ranges from 64,643 to 106,895 depending on the minimum dwell constraint. Both algorithms indicate that the minimum dwell policy must be close to 1:1.6 in order to achieve an average ratio of 1:2 while the minimum dwell policy must be at or near 1:2.6 in order to achieve an average ratio of 1:3.

7.3 Demand scenarios

The demand scenario which drives the parameters for demand by location and month is a critical component to this analysis. Clearly, different demand scenarios will produce different solutions. In this section we evaluate some different generated scenarios other than the historical sce-

Table 5 Locations/unit for deployment lengths and minimum BOG:Dwell ratios

d*	Vertices	Edges	First-fit	First-fit swap		FFL Swap	
			Loc/unit	Loc/unit	Diff(%)	Loc/unit	Diff(%)
Min Dwell = 1:1							
9	693	90,706	2.3495	1.6068	−31.61	1.3493†	−42.57
10	642	80,806	2.1762	1.4667	−32.60	1.2661†	−41.82
11	590	74,246	1.9853	1.5098	−23.95	1.2207†	−38.51
12	507	64,643	1.9034	1.2705	−33.25	1.2358†	−35.07
13	468	59,949	1.8373	1.2440	−32.29	1.1542	−37.18
14	436	55,297	1.7670	1.1690	−33.84	1.1343	−35.81
15	406	49,098	1.7240	1.2448	−27.80	1.1876	−31.11
Min Dwell = 1:2							
9	693	132,466	1.9178	1.2877	−32.86	1.2162†	−36.58
10	642	122,454	1.7710	1.2660	−28.51	1.1377	−35.76
11	590	112,071	1.6254	1.2367	−23.91	1.0686	−34.26
12	507	88,439	1.5436	1.1533	−25.29	1.0814	−29.94
13	468	80,439	1.4276	1.1201	−21.54	1.0912	−23.57
14	436	74,045	1.4051	1.1533	−17.92	1.0845	−22.82
15	406	66,159	1.4819	1.2329	−16.80	1.0899	−26.45
Min Dwell = 1:3							
9	693	167,920	1.6043	1.1626	−27.53	1.0801	−32.68
10	642	153,590	1.5253	1.1520	−24.47	1.0483	−31.27
11	590	137,213	1.4282	1.1690	−18.15	1.0423	−27.02
12	507	106,895	1.3020	1.1026	−15.31	1.0618	−18.45
13	468	95,272	1.3120	1.0729	−18.22	1.0670	−18.67
14	436	87,664	1.2426	1.0947	−11.90	1.0885	−12.40
15	406	77,219	1.1554	1.0293	−10.91	1.0293	−10.91

* Deployment length in months
† No improvement to the FFL solution

nario. We also acknowledge the use of steady-state demand scenarios, used for theoretical planning and the absence of reasonable demand forecasting mechanism, and demonstrate the expected consistency in results when coloring a steady-state deployment schedule.

7.3.1 Demand excursion

Alternate demand scenarios are introduced in order to determine if the performance of the coloring algorithms and heuristics is attributable to the characteristics of the historical demand input. Recalling that the historical demand is not evenly distributed by location, our initial excursion implements a similar aggregate demand pattern in which demand is evenly distributed among the eight locations. Over time, the excursion demand is not steady-state and is depicted in Fig. 9.

With the results from the excursion, we are interested in determining if some of the early trends still hold, namely if FFL continue to produce better solutions in terms of location than first-fit and if swapping heuristics improve the solution. Also of interest is any change to relationship between location improved solutions and dwell ratio measurements. Scheduling solutions are produced with each method while varying the deployment length parameter and minimum dwell constraint.

Table 7 shows the results of using a BOG:Dwell constraint of 1:1 and deployment lengths of 9, 12 and 15 months. The number of units required is the same for all four assignment methods. Interestingly, the number of units is 193 for both the 9- and 12-month deployment solutions. With 15-month deployments, the number of units decreases to 176. FFL Swap is clearly capable of more successfully minimizing the number of locations per units when demand is more evenly distributed, regardless of deployment length. First-fit swap tends to consistently produces slightly better average dwell results, but the difference from other methods is relatively insignificant.

When the BOG:Dwell ratio constraint is tightened to 1:2, the general observations remain the same, as seen in Table 8. Of course, the number of units required for the scheduling solutions increases; however, the requirement of 273 units

Table 6 Average Dwell for deployment lengths and minimum BOG:Dwell ratios

d*	Vertices	Edges	First-fit	First-fit swap		FFL		FFL swap	
			Avg Dwell	Avg Dwell	Diff(%)	Avg Dwell	Diff(%)	Avg Dwell	Diff(%)
Min Dwell = 1:1									
9	693	90,706	1.2046	1.2275	1.90	1.3179	9.41	1.3935	**15.68**
10	642	80,806	1.1315	1.1376	0.54	1.2683	12.09	1.3055	**15.38**
11	590	74,246	1.1965	1.1980	0.13	1.3667	14.22	1.3938	**16.49**
12	507	64,643	1.3039	1.3478	3.37	1.4119	8.28	1.4361	**10.14**
13	468	59,949	1.2511	1.3837	10.60	1.3543	8.25	1.3911	**11.19**
14	436	55,297	1.2514	1.4038	**12.18**	1.3539	8.19	1.4001	11.88
15	406	49,098	1.1579	1.2333	6.51	1.2636	9.13	1.2796	**10.51**
Min Dwell = 1:2									
9	693	132,466	2.2860	2.3091	1.01	2.3987	**4.93**	2.3987	4.93
10	642	122,454	2.1373	2.1893	2.43	2.3417	9.56	2.3436	**9.65**
11	590	112,071	2.1872	2.2519	2.96	2.4531	**12.16**	2.4321	11.20
12	507	88,439	2.2940	2.4348	6.14	2.4978	**8.88**	2.4949	8.76
13	468	80,439	2.1978	2.3946	**8.95**	2.3930	8.88	2.3926	8.86
14	436	74,045	2.1737	2.3034	5.97	2.4107	**10.90**	2.3882	9.87
15	406	66,159	2.1520	2.1563	0.20	2.3453	**8.98**	2.3209	7.85
Min Dwell = 1:3									
9	693	167,920	3.3099	3.4129	3.11	3.5285	**6.60**	3.5199	6.34
10	642	153,590	3.1643	3.2700	3.34	3.4101	**7.77**	3.3685	6.45
11	590	137,213	3.1799	3.2317	1.63	3.4717	**9.18**	3.4207	7.57
12	507	106,895	3.2943	3.4161	3.70	3.4327	**4.20**	3.4238	3.93
13	468	95,272	3.1458	3.2849	**4.42**	3.2732	4.05	3.2689	3.91
14	436	87,664	3.2784	3.2456	−1.00	3.2931	**0.45**	3.2666	−0.36
15	406	77,219	3.4185	3.3251	−2.73	3.4185	0.00	3.3262	−2.70

* Deployment length in months
The bold font signifies which method achieved the best result (greatest positive diff %) for each scenario

remains equal for all 9 and 12-month solutions. The 15-month solutions require 240 units. FFL Swap continues to produce perfect solutions when measured against the location objective, while the swapping heuristic improves the First-Fit solution to a nearly optimal location solution. First-fit still dominates the average dwell statistic on a relatively insignificant scale.

7.3.2 *Steady-state demand*

Often, steady-state demand scenarios are used for theoretical planning or in the absence of demand forecasts. Naturally, the number of units required to fulfill a steady-state scenario will change if minimum dwell policies are adjusted. For the steady-state computations, we generated a demand scenario in which demand remains constant for 64 total units. Demand requirements are distributed unevenly by location but remain constant over time in each location.

The results show that changes in deployment lengths have no impact on the number of units required when demand represents a steady-state scenario. As the deployment length increases, the number of deployments decreases; however, the number of units remains constant. The solution for all algorithms assigns every unit to only one location. Moreover, the minimum, maximum and average dwell statistics are always equal to and bound by the minimum dwell policy constraint. In fact, the steady-state solutions are the same regardless of the distribution by location.

While steady-state demand is useful for theoretical planning to address supply and demand capabilities, it is not informative for the DSUA problem in practice. This is due to the nature of persistent conflict, which historically does not adhere to steady-state requirements. Even an unlikely partial steady-state scenario is easily disrupted by surges to the rotation pattern, making a heuristic-based approach relevant.

8 Conclusions

We demonstrate how the DSUA problem can be efficiently structured and solved using modifications to existing graph

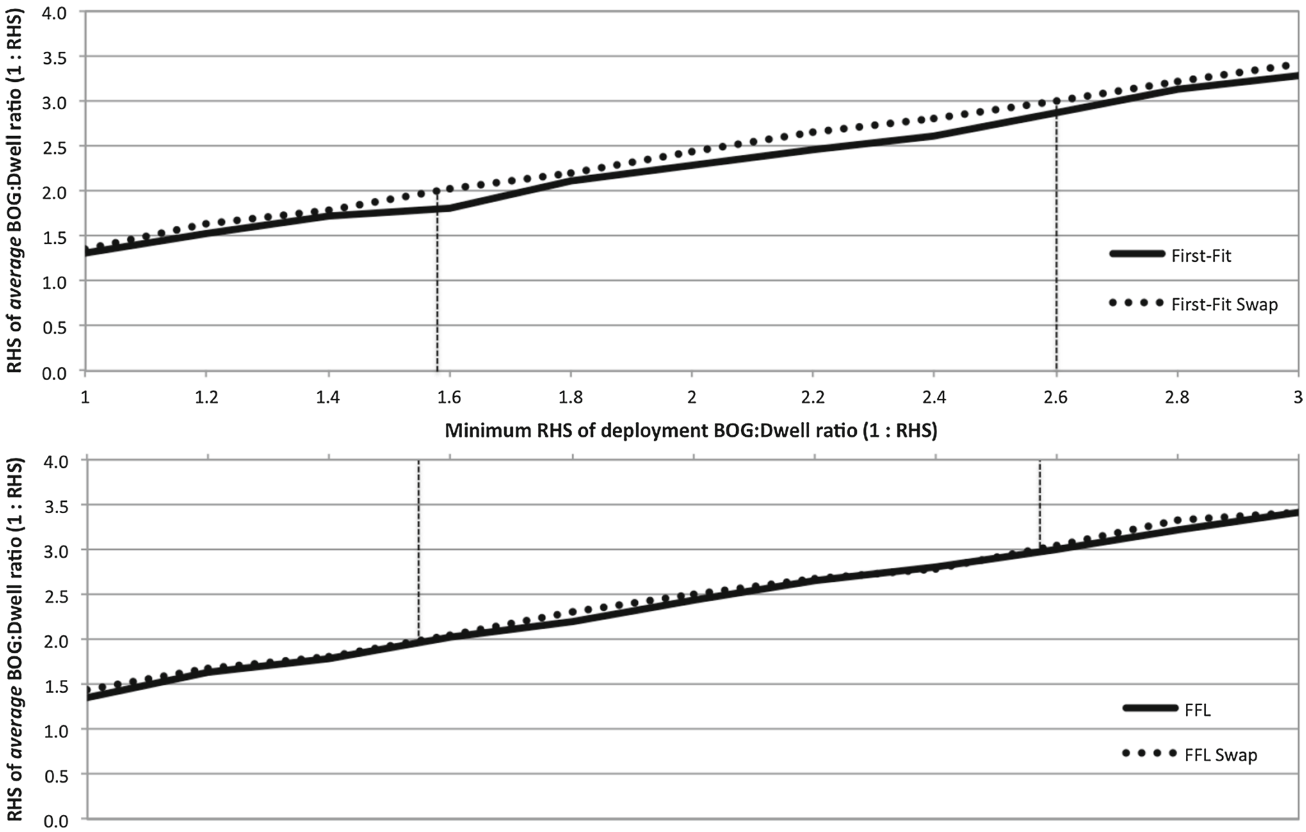

Fig. 8 Average BOG:Dwell ratios with changes to the dwell policy constraint ($V = 507$)

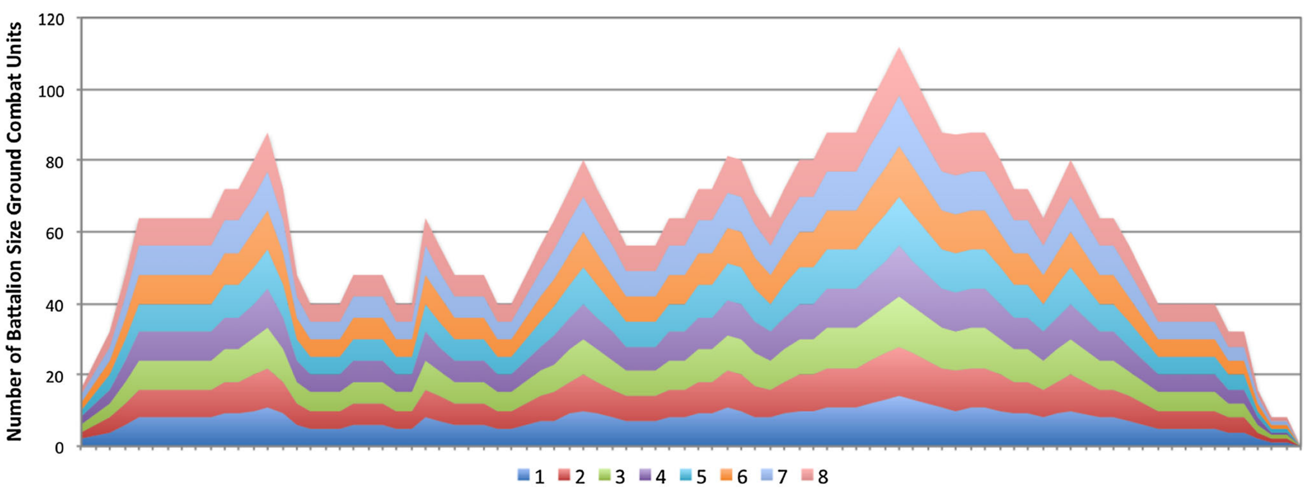

Fig. 9 Demand excursion of 460 deployments evenly distributed by location ($T = 86$)

coloring techniques. Unlike traditional coloring problems, the DSUA problem seeks to improve multiple objectives. Section 4 proves the theoretical limitations of BOG:Dwell in steady-state scenarios relative to the number of units available for sourcing and deployment length.

In Sect. 5, we show that the DSUA problem can be formulated as an integer programming problem. Due to the special characteristics of the DSUA problem in which we can assume fixed deployment lengths and the set of variable and constraints is extremely large, graph coloring methods reduce the solution complexity and provide a more efficient method for targeting multiple objectives at once. Moreover, the scheduling structure, which builds upon an indexed time horizon lends itself to algorithms that are proven to optimize the coloring solutions of interval graphs in respect to the minimum number of colors.

Table 7 Demand excursion with $d := \{9, 12, 15\}$ and $w = d$

Method	Locations per unit			Max locations			Average Dwell			w
	9	12	15	9	12	15	9	12	15	
First-fit	2.6373	2.6373	1.8296	5	4	3	1.2245	1.2926	1.1510	1:1
First-fit swap	1.1762	1.1762	1.0511	3	2	3	**1.2428**	**1.3137**	**1.1902**	
FFL	1.0518	1.0518	1	2	1	1	1.2238	1.2926	1.1510	
FFL swap	1	1	1	1	1	1	1.2264	1.2926	1.1510	

The bold values signifies which method achieved the best result (greatest positive diff %) for each scenario

Table 8 Demand excursion with $d := \{9, 12, 15\}$ and $w = 2d$

Method	Locations per unit			Max locations			Average Dwell			w
	9	12	15	9	12	15	9	12	15	
First-fit	1.9707	1.6117	1.2208	3	3	2	2.3279	2.3300	2.0843	1:2
First-fit swap	1.0256	1	1	3	1	1	**2.3581**	**2.3964**	**2.1260**	
FFL	1	1	1	1	1	1	2.3279	2.3300	2.0843	
FFL swap	1	1	1	1	1	1	2.3279	2.3300	2.0843	

The bold values signifies which method achieved the best result (greatest positive diff %) for each scenario

Two primary algorithms, first-fit and FFL, are presented along with a swapping heuristic which seeks to improve their solutions against the objective which minimizes locations per unit. Solutions are also evaluated based on BOG:Dwell performance, which while not targeted by the algorithms, seems to perform in concert with the location objective. Perhaps most importantly, we show that a deterioration in BOG:Dwell ratio performance is not sacrifice for improvements with the location objective. Naturally, BOG:Dwell conflicts with an objective to minimize units because an increase in the supply for unit sourcing can be used to increase average dwell time.

Using a historical demand scenario and other demand scenario excursions, computational experiments demonstrate the effectiveness of First-fit swap and FFL swap in producing DSUA problem solutions that are optimal in respect to the unit objective. Depending on the complexity of the demand scenario and policy constraints, solutions are reasonably close to a goal of one location per unit and always average less than two locations per unit. FFL swap is the most effective method to target the location objective and first-fit swap always provides an optimal solution against the unit supply objective while significantly targeting the location objective. No method provides a significant advantage over another in terms of BOG:Dwell ratios; however, the results from tightening the minimum BOG:Dwell policy constraint provide an expectation for average BOG:Dwell results that is important for consideration when balancing the implication of budget decision with the size of the sourcing pool with statistical goals such as average BOG:Dwell.

The DSUA problem is of significant importance to the Army from multiple perspectives. Clearly, dwell and location performance measures that are indicators of stress on the force and operational consistency, which can be used to drive palatable DSUA solutions and influence policies. For example, since individuals do not align with single units throughout their careers, we cannot expect individuals dwell policies to meet or exceed unit capabilities. Moreover, the ability to minimize the number of units required to meet expected demand has even broader and more direct budget and force structure implications. In an era of tightening budget constraints, the ability to meet demand with a minimum size of force is of great importance. This research demonstrates how the Army can use these methods to minimize the size of its force structure requirements and reduce costs while simultaneously understanding the impact in regards to other measures that affect operational readiness and individual experiences that influence retention.

An important component to this research is the ability to use forecasted demand to drive DSUA problem solutions. Because of the uncertainty of demand, the time horizon for a reliable solution may be limited. Future research of these methods could provide alternative methods that would account for additional uncertainty in demand by incorporating theory from online interval graphs; therefore, producing a solution for a shorter time interval before demand further along the time horizon is observed.

References

Alfares, H. (2004). Survey, categorization, and comparison of recent tour scheduling literature. *Annals of Operations Research*, *127*, 1–4.

Aviles, S. M. (1995). *Scheduling Army Deployment to Two Nearly Simultaneous Major Regional Conflicts*. Monterey: Naval Postgraduate School.

Baker, K. R. (1976). Workforce allocation in cyclical scheduling problems: a survey. *Operational Research Quarterly*, *27*, 155–167.

Blochliger, I. (2004). Scheduling; staff; modeling tutorial, modeling staff scheduling problems. A tutorial. *European Journal of Operational Research*, *158*, 533–542.

Bonds, T. M., Baiocchi, D., & McDonald, L. L. (2010). *Army Deployments to OIF and OEF*. Santa Monica: RAND Corporation.

Cazals, F., & Karande, C. (2008). A note on the problem of reporting maximal cliques. *Journal of Theoretical Computer Science*, *407*, 564–568.

Cheng, T. C. E., & Chen, Z.-L. (1994). Parallel-machine scheduling problems with earliness and tardiness penalties. *Journal of the Operational Research Society*, *645*, 685–695.

Costa, D., Hertz, A., & Dubuis, C. (1995). Embedding a sequential procedure within an evolutionary algorithm for coloring problems in graphs. *Journal of Heuristics*, *1*, 105–128.

Dabkowski, M., Kwinn, M. J., Miller, K., & Zais, M. (2009). Unit BOG: Dwell...a closed-form approach. *Phalanx*, 42(4), 11–14.

Department of the Army, AR 525–29: Army Force Generation, March 2011.

Department of the Army, Army Deployment Period Policy, August 2011.

Department of the Army, FM 3-24: Counterinsurgency, December 2006.

Department of the Army, Personnel Policy Guidance for Overseas Contingency Operations, July 2009.

Galinier, P., & Hertz, A. (2006). A survey of local search methods for graph coloring. *Computers and Operations Research*, *33*, 2547–2562.

Gamach, M., Hertz, A., & Ouellet, J. O. (2007). A graph coloring model for a feasibility problem in monthly crew scheduling with preferential bidding. *Computers and Operations Research*, *34*, 2384–2395.

Glover, F., & McMillan, C. (1986). The general employee scheduling problem: an integration of MS and AI. *Computers and Operations Research*, *13*, 563–573.

Glover, F. (1989). Tabu search - part I. *ORSA Journal of Computing*, *1*, 190–206.

Graham, R.L., Lawler, E.L., Lenstra, J.K., & Rinnooy Kan, A.H.G. (1979). Annals of Discrete Mathematics 5: Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey, Hammer, P.L. and Johnson, E.L. and Korte, B.H.. North-Holland Publishing Company.

Hodgson, T. J., Melendez, B., Thoney, K. A., & Trainor, T. (2004). The deployment scheduling anayisis tool (DSAT). *Mathematical and Computer Modelling*, *39*, 905–924.

Hughes, David W., Zais, Mark M., Kucik, Paul, & Huerta, Fernando M. (2011). ARFORGEN BOG: Dwell Simulation. *Operations Research Center of Excellence*.

Kierstead, H. A. (1988). The linearity of first-fit coloring of interval graphs. *Society for Industrial and Applied Mathematics*, *1*, 526–530.

Kierstead, H. A., & Qin, J. (1995). Coloring interval graphs with first-fit. *Discrete Mathematics*, *144*, 47–57.

Kilcullen, D. (2006). Twenty-eight articles: fundamentals of company-level counterinsurgency. *Military Review*, *86*, 50.

Leighton, F. T. (1979). A graph coloring algorithm for large scheduling problems. *Journal of Research of the National Bureau of Standards*, *84*(6), 489–506.

Lenstra, J. K., Rinnooy Kan, A. H. G., & Brucker, P. (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, *1*, 343–362.

Malaguti, E., & Toth, P. (2010). A survey on vertex coloring problems. *International Transactions in Operational Research*, *17*, 1–34.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural Multidisciplinary Optimization*, *26*, 369–395.

McKinzie, K., & Barnes, J. W. (2004). A review of strategic mobility models supporting the defense transportation system. *Mathematical and Computer Modeling*, *39*, 839–868.

Palubeckis, G. (2008). On the recursive largest first algorithm for graph colouring. *International Journal of Computer Mathematics*, *85*, 191–200.

Reed, Heather (2011). *Wartime Sourcing: Building Capability and Predictability through Continuity*. Military Review, May-June 2011..

Rosen, K. H. (2011). *Elementary Number Theory and Its Applications* (6th ed.). Boston: Addison Wesley Longman.

Van den Bergh, J., Beliën, J., De Brueker, P., & Demeulemeester, E. (2013). Personnel scheduling: a literature review. *European Journal of Operational Research*, *226*, 367–385.