Efficient Solution Methods for the Cumulative-Interference Channel Assignment Problem Using Integer Optimization and Constraint Programming

Paul J. Nicholas[1,2](✉) and Karla L. Hoffman[2]

[1] Johns Hopkins University Applied Physics Laboratory, Laurel, MD 20723, USA
paul.nicholas@jhuapl.edu
[2] George Mason University, Fairfax, VA 22030, USA
khoffman@gmu.edu

Abstract. Interest in the channel assignment problem (CAP) has been growing rapidly with both the spread of wireless data networks and the increasing scarcity of electromagnetic (EM) spectrum. The ability to efficiently reuse available EM channels is heavily dependent on co-channel interference, i.e., interference occurring between two radios using the same channel but not communicating on the same network. The vast majority of CAP research considers only the interference between any pair of radios, but many radio systems – including the mobile ad-hoc networks we consider – are sensitive to the effects of cumulative interference. In previous work, we describe the vast computational challenges of considering cumulative interference within a CAP. We present a new method to solve this problem via heuristics, integer optimization, and constraint programming techniques. We apply our methods to realistic data sets from a large U.S. Marine Corps operational scenario and provide detailed performance results. To our knowledge, we are the first to describe algorithms for solving realistic, large-scale cumulative-interference minimum-order and minimum-cost channel assignment problems to global or near-global optimality.

Keywords: Constraint programming · Integer optimization · Channel assignment problem · Electromagnetic spectrum

1 Introduction

Interest in the *channel assignment problem* (*CAP*) has been growing rapidly with both the spread of wireless data networks and the increasing scarcity of

Partial support for this research was provided by the Office of Naval Research grant N00014-15-1-2176.

L.-M. Rousseau and K. Stergiou (Eds.): CPAIOR 2019, LNCS 11494, pp. 443–460, 2019.
https://doi.org/10.1007/978-3-030-19212-9_29

electromagnetic (EM) spectrum [1,13,35]. Efficient channel allocation schemes leverage *channel reuse*, where a channel is a contiguous block of EM spectrum. The ability to reuse a channel is dependent on (among other things) *co-channel interference*, i.e., interference occurring between two radios assigned the same channel but that do not wish to communicate. The vast majority of research on the CAP considers only pairwise interference constraints [1] due to the computational challenges of explicitly representing cumulative interference, and the ease with which the problem can be represented as a *graph-coloring problem* [4,9,25,27,38]. This seemingly simple problem is *NP-complete* [6], and yet the realistic cumulative interference constraints we model are much more difficult [26–28].

We consider the challenge of a *spectrum manager* who must determine an efficient channel allocation scheme to support radio communications over a certain period of time for mobile units operating on rough terrain. We specifically consider the use of wideband *mobile ad-hoc network* (*MANET*) radios fielded by the United States Marine Corps (USMC), but our approach generalizes to other military services and any EM transceiver system requiring a discrete channel assignment. The spectrum manager knows the capabilities of each radio and their starting locations, and has a rough understanding of their future locations within the *operating area.* Using this information and terrain elevation data, the spectrum manager must determine the minimum number of channels required to support communications with an acceptable level of co-channel interference. Further, since each radio requires manual assignment, the spectrum manager is responsible for the reallocation of channels whenever the situation changes, and therefore desires to minimize the number of channel changes over time.

Due to the computational difficulties of exactly solving the CAP, heuristics are often used to solve the problem [1,23]. While heuristics may provide usable solutions in reasonable amounts of time, we feel that optimality bounds are important for understanding the goodness of a particular solution, especially since spectrum is increasingly crowded and scarce, and communications may be critical to the success of a military operation.

Dunkin et al. [9] describe the challenge of using cumulative interference constraints, and instead use simple binary and tertiary constraints (i.e., groups of three interfering radios) using a *constraint satisfaction* approach. Daniels et al. [7] formulate an integer CAP that considers cumulative interference and establish the NP-hardness of the problem. Fischetti et al. [11] use pre-processing and *branch-and-cut* to solve their cumulative interference CAP, but their problem sizes are much smaller than those studied here and they consider relatively few sources of interference, i.e., they have a relatively small number of constraints.

We use integer and constraint programming methods to develop more efficient methods of channel allocation. Our first problem minimizes the number of required channels (i.e., *minimum order*), subject to cumulative co-channel interference constraints for any given instance in time, and the second problem minimizes the number of channel assignment changes over time (i.e., *minimum cost*). We use realistic radio performance data from large-scale, high-fidelity simulations of U.S. Marine

Corps operational scenarios (the data are available to the research community at [12]). To our knowledge, we are the first to solve to global- or near-global optimality the minimum-order and minimum-cost channel assignment problems for large, realistic datasets while also considering the effects of cumulative co-channel interference and the costs of manual channel changes. We believe that the tools created for this application are likely to be appropriate for other complex graph-coloring problems, as well.

This paper is organized as follows. Section 2 provides an overview of our model of MANET operations. Section 3 describes our formulation and computational results in solving the minimum-order channel assignment problem, and Sect. 4 does the same for the minimum-cost CAP. Section 5 provides conclusions and suggestions for future research.

2 Model of MANET Communications

We create a network model to simulate the key aspects of a MANET formed by tactical wideband radios at a given moment in time (i.e., time step). Let $r \in R$ (alias s) represent each MANET radio. Each radio is permanently assigned to a MANET *unit* $u \in U$, indicated by the set of *logical arcs* $(r, u) \in L$. In a military scenario, a unit may represent a tactical military organization, such as an infantry company or battalion headquarters. Let the set of *nodes* N (indexed by n) consist of both radios R and units U, i.e., $n \in N = R \cup U$. Let a channel $c \in C$ be a contiguous range of EM frequencies, where C is the set of available *orthogonal* (i.e., non-interfering) channels. Each unit u and the radios assigned to it require a channel assignment.

Let $(r, s) \in W$ indicate the set of arcs representing wireless transmissions between all radios $r, s \in R$. A unit $u \in U$ forms a separate MANET among its assigned radios using the available wireless arcs $(r, s) \in W : (r, u) \in L, (s, u) \in L$. Figure 1 shows two separate units (indicated in blue and green) and their assigned radios. The solid lines indicate bidirectional wireless arcs $(r, s) \in W$ between radios. Any radio (e.g., radio r in Fig. 1) communicates with its *network control radio* (e.g., radio s) via these arcs (a radio may route through other radios in the same unit to reach the network control radio). All radios are subject to co-channel interference from any other radios assigned to different units but operating on the same channel and geographically close enough to cause interference; this is indicated by dashed gray arrows directed to r (other lines withheld for clarity).

To calculate both co-channel interference and the strength of desired wireless transmissions between intra-unit radios, we calculate the *received signal strength* (*RSS*) along all wireless arcs $(r, s) \in W$ in dBm (power ratio in decibels relative to milliwatts). We instantiate our scenarios in Systems Toolkit (STK) [3] and then use Python and the Terrain Integrated Rough Earth Model (TIREM) of Alion Science & Technology Corporation [2] to calculate path loss considering the technical specifications of each radio and the effects of terrain, atmospheric absorption, etc.

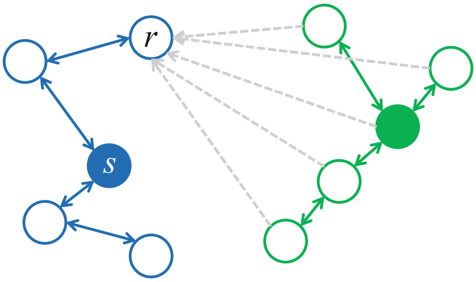

Fig. 1. Simple example of two units (indicated in blue and green) with network control radios (solid circles) and other radios (open circles). Wireless arcs are indicated by arrows. The radios within each unit must be capable of bi-directional communication with their unit's network control radio via direct communication or routing through other radios in the same network. All radios are subject to co-channel interference (dashed arrows) from other radios assigned to different units but operating on the same channel. (Color figure online)

For each radio $s \in R$, we follow [1] and pre-calculate the maximum allowable interference in watts $max\_interference_s^c$. This calculation is based on the RSS between radios and each particular radio's required *signal-to-interference ratio* (SIR), a measure of signal quality [33]. Any co-channel interference above this level severs the shortest path and thus disconnects the radio from its assigned network control radio. Among radios not assigned to the same unit but operating on the same channel, the RSS represents co-channel interference. The magnitude of co-channel interference along all arcs $(r, s) \in W$ for each available channel $c \in C$ is pre-calculated in watts, and is indicated by $interference_{rs}^c$.

We use realistic datasets generated from high-fidelity simulations of U.S. Marine Corps operations. We find the largest scenario, depicting a Marine Expeditionary Force (MEF) of 60,000 Marines conducting a large amphibious operation and based on Integrated Security Construct B [8], to be the most computationally interesting. We generate separate datasets at 20 different *time steps* (i.e., discrete moments) within the scenario (each containing the locations of 118 units comprising 1887 total radios). See [26] for full details of our scenarios.

3 Minimum-Order Channel Assignment Problem (MO-CAP)

3.1 Problem Formulation

The MO-CAP aims to minimize the total number of channels required to support MANET operations at a given moment in time.

Variables:

$$X_n^c = \begin{cases} 1, \text{if node } n \text{ uses channel } c \\ 0, \text{otherwise} \end{cases} \qquad \forall n \in N, c \in C \tag{1}$$

$$Y^c = \begin{cases} 1, \text{if channel } c \text{ is used} \\ 0, \text{otherwise} \end{cases} \qquad \forall c \in C \tag{2}$$

Constraints:
We enforce the definition of Y^c via:

$$X_u^c \leq Y^c \quad \forall u \in U, c \in C. \tag{3}$$

Each radio is assigned the same channel as its associated unit:

$$X_r^c = X_u^c \quad \forall c \in C, (r, u) \in L. \tag{4}$$

To ensure each unit u is assigned one and only one channel, we add the constraints:

$$\sum_{c \in C} X_u^c = 1 \quad \forall u \in U. \tag{5}$$

Two radios from different units are subject to interference if they are both assigned to the same channel. This assignment will only be allowed if the received $interference_{rs}^c$ between these two radios is less than the precalculated allowable total interference, $max\_interference_s^c$. One way of representing this pairwise interference is:

$$interference_{rs}^c X_r^c X_s^c \leq max\_interference_s^c \quad \forall (r, s) \in W, c \in C. \tag{6}$$

To model the total aggregate interference that a radio receives, we follow the lead of [19], and assume the cumulative effects of jamming sources on the same channel are additive (in watts). That is, a radio $s \in R$ may be unable to use a channel $c \in C$ because the total sum of interference exceeds the threshold $max\_interference_s^c$, even if the interference received from any single radio is less than the threshold. Summing along all arcs yields:

$$\sum_{r:(r,s) \in W} interference_{rs}^c X_r^c X_s^c \leq max\_interference_s^c \quad \forall s \in R, c \in C. \tag{7}$$

To linearize these constraints, we introduce the binary variable Z_{rs}^c where:

$$Z_{rs}^c = \begin{cases} 1, \text{if } X_r^c = X_s^c = 1 \\ 0, \text{otherwise} \end{cases} \qquad \forall r, s \in R, c \in C \tag{8}$$

which is enforced via:

$$Z_{rs}^c \geq X_r^c + X_s^c - 1 \qquad \forall r, s \in R, c \in C \tag{9}$$

$$Z_{rs}^c \leq X_r^c \qquad \forall r, s \in R, c \in C \tag{10}$$

$$Z_{rs}^c \leq X_s^c \qquad \forall r, s \in R, c \in C. \tag{11}$$

Our cumulative co-channel interference constraints are thus represented:

$$\sum_{r:(r,s)\in W} interference_{rs}^{c} Z_{rs}^{c} \le max\_interference_{s}^{c} \quad \forall s \in R, c \in C. \tag{12}$$

Given the results of radio propagation simulation in a military scenario, we precalculate the $max\_interference_{s}^{c}$ values (using the method described above), and fix the assignment of radios to their respective units (indicated by arcs $(r,u) \in L$).

Since the goal is to minimize the total number of channels required, our objective function is:

$$\min \sum_{c\in C} Y^{c}. \tag{13}$$

3.2 Computational Challenges

The MO-CAP is relatively easy to understand and describe. However, it suffers from several serious computational difficulties when the full problem is simply provided to a commercial solver (e.g., CPLEX or Gurobi) with our realistic datasets. First, commercial solvers may be sensitive to vast differences in input parameters. In our simulated datasets, our interference values vary by 24 orders of magnitude, and are generally quite small. Also, non-integral input data may result in highly *fractionalized* LP solutions, as the solver will attempt to pack the most units (including fractions of units) onto the same channel.

Another computational problem (also observed by [32]) is that of *symmetry*, which occurs when channel assignments may be changed among units with no corresponding change in the objective function value [24]. The very *near* symmetry that is characteristic of our datasets (as opposed to *exact* symmetry) results from some units being located near each other, and is especially difficult for solvers to detect and mitigate [31].

Some of these computational problems could be avoided if we considered only pairwise interference constraints, as IP and constraint satisfaction solvers reformulate these pairwise constraints into clique constraints and then handle these structures very efficiently. Unfortunately, these constraints alone do not adequately represent our real-world problem, and will cause at least a few radios to be disconnected from their respective MANETs.

A simple "brute force" IP method (i.e., using CPLEX to solve the full problem as-is, without providing any initial solution or conducting preprocessing) fails to obtain useful answers to the Marine Corps scenario, even after two weeks of computation on a cluster of 14 high-performance desktop computers. In an attempt to improve the solution process, we create a simple greedy heuristic that iteratively "packs" units onto channels until the channel is "full," and then starts with the next channel. We provide the heuristic solution as an initial feasible solution to CPLEX and attempt to solve the problem for a single time step. We find that after 60 h of runtime, CPLEX improves upon the initial feasible solution, but the obtained solution has an optimality gap of 77%. This indicates that our heuristic may not be finding very good solutions and/or the lower bounding technique is not very effective, and that we require more sophisticated methods if we are going to solve realistic instances of this problem with certifiably-good solutions.

3.3 Integer Programming Solution Method and Results

Rather than simply "throw" the computationally-challenging cumulative interference constraints (12) at a solver, we preprocess the constraints to create simplified and more computationally tractable *packing constraints*. For example, suppose two specific nodes r and s (not assigned to the same unit) are not both allowed to be assigned to channel c because to do so would violate the associated interference constraint. This may be represented as:

$$X_r^c + X_s^c \leq 1. \tag{14}$$

We use Python and the `mpmath` library [18], which allows the use of arbitrary-precision floating point mathematics, to identify unacceptable pairs of radios and handle the extremely small interference values present in our realistic data sets.

To generalize for larger n-tuples of units above pairs (triplets, quadruplets, etc.), let $S \subset U$ be a subset of units that cannot all be assigned to the same channel c. We can represent such a restriction of assignments as:

$$\sum_{r \in S} X_r^c \leq |S| - 1. \tag{15}$$

Preprocessing all such unacceptable combinations and adding them as constraints would effectively replace the cumulative co-channel interference constraints (12). However, identifying all combinations would be computationally prohibitive (as they grow exponentially with both the number of units and available channels) and unnecessary, as many combinations will be redundant and/or represent negligible levels of co-channel interference.

Instead, we dynamically add these higher-order constraints to the formulation only as needed via *lazy constraints*, which are constraints that are checked for violation whenever an integer solution to the current formulation is found. They are added on an as-needed basis [17]. This approach avoids the problem of very small numbers in CPLEX, as we can process the constraints outside of the solver (e.g., in Python), and then add the much-simplified packing constraints (15) dynamically. Also, since the solver is no longer required to calculate cumulative interference at each radio, the formulation no longer requires the index $r \in R$. That is, we are now concerned only with the cumulative interference received at each unit. By removing the index $r \in R$, we reduce the number of decision variables in the problem by an order of magnitude.

After building an initial problem instance with pairwise constraints using Python and Pyomo [15], we send the problem to CPLEX via the Python API and indicate to the solver that we wish to initiate lazy constraint *callbacks*. Upon

Table 1. MO-CAP results by time step using pairwise and lazy constraints. "Time" indicates the time at which the displayed solution and optimality gap is obtained, during a total runtime of 9,000 s.

Time Step	Number lazy constraints	Highest- order lazy constraint	Solution value	Time (s)	Gap	Improvement over heuristic
1	49	5	46	1356.53	2.17%	9.80%
2	25	5	37	1333.92	0%	22.92%
3	87	6	36	4432.53	5.56%	21.74%
4	62	5	34	7828.09	5.88%	27.66%
5	9	5	33	678.23	0%	23.26%
6	104	6	36	4086.04	2.78%	29.41%
7	67	5	37	1737.45	0%	24.49%
8	57	5	31	8614.79	6.45%	26.19%
9	21	8	32	271.19	0%	25.58%
10	0	0	34	248.16	0%	30.61%
11	121	11	33	5997.82	3.03%	26.67%
12	29	5	36	927.38	2.78%	16.28%
13	104	6	32	2510.22	3.12%	25.58%
14	69	6	31	1780.48	3.23%	27.91%
15	147	6	38	1669.23	2.63%	22.45%
16	119	8	36	4194.86	5.56%	23.40%
17	128	6	37	3092.38	2.70%	24.49%
18	8	4	31	245.20	0%	22.50%
19	99	5	30	1673.56	3.33%	25.00%
20	47	5	37	1268.30	0%	24.49%
Aver	**67.6**	**5.6**	**34.9**	**2697.32**	**2.46%**	**24.02%**

finding an integer solution that is feasible with the current constraints, the solver runs our lazy constraint callback code. The code checks the feasibility of the current solution in the full problem, i.e., it checks if the solution satisfies each of the constraints (12). This can be calculated in polynomial time, specifically $\mathcal{O}\left(|R|^2|C|\right)$. If infeasibility exists, we add the lowest-order constraint (15) to the constraint set to prevent the same units from being assigned the same channel again. CPLEX then continues the search process with these new constraints added into the formulation. The process repeats until optimality is achieved or a time limit is reached.

Table 1 displays results for each time step in the Marine Corps scenario, including the number of lazy constraints (and the order of the highest-order lazy constraint), and solution results. Each time step is run for 9,000 s, or until optimality is obtained. The times in Table 1 indicate the time when the displayed solution value and optimality gap is obtained; those time steps with a non-zero optimality gap fail to converge within 9,000 s. Our results are obtained using a Dell Mobile Precision 6800 laptop with 32 GB of RAM and an Intel Core i7-4940MX processor running at 3.1 GHz. We use IBM ILOG CPLEX version 12.6.2 and Python 2.7.

The lazy constraint approach to solving the MO-CAP yields results far superior to our previous methods. The solutions are on average 24% lower than the heuristic, and each solution has an associated optimality gap. On seven time steps, optimality is achieved. Even for those for which optimality is not proven, the method finds solutions within one or two channels of optimality.

We next improve on our lazy constraint method by adding the constraints specifying the *maximum clique*, which is the largest *complete sub-graph* formed from among the pairwise interference constraints. We use the `NetworkX` Python library [14] to find the maximum clique, which relies on the algorithm of [5] as adapted by [36]. Let $M \subset U$ be the subset of units in the maximum clique. The maximum clique constraint takes the form:

$$\sum_{u \in M} X_u^c \leq 1 \quad \forall c \in C. \tag{16}$$

That is, only one unit in the clique may be assigned any given channel. Adding this constraint forces the lower bound up significantly and allows the optimization engine to search in a much smaller feasible region. After we add the maximum clique, we then add all remaining pairwise constraints that are not included in this clique constraint. The lazy constraint method is used again to generate any higher-order interference constraints.

The results of this method are displayed in Table 2, where bolded values indicate an improvement over the previously-described technique. Again, each time step is run for 9,000 s, or until optimality is obtained, and "Time" indicates solver time when the displayed solution value and optimality gap is obtained. Overall, inclusion of the maximum clique reduces average runtime to obtain solutions within one channel of optimality. For the problem associated with time step 3, this method obtains a solution that requires one less channel than that identified without use of the maximum clique. On eight time steps, this method reduces the known optimality gap, and on 12 time steps, the method obtains the provably-optimal solution (five more time steps than the previous method). It is interesting to note that the size of the maximum clique (which itself provides a lower bound on the number of required channels) is within one of the best-known solution for each time step. This is indicative of the power of the maximum clique constraint. We note that there is a clique constraint generator within CPLEX, but this procedure does not find this very strong clique; the overall solution times obtained when the maximum clique constraint is removed and CPLEX clique generator is turned on to aggressive yields results no better than those obtained with default parameters for CPLEX.

We also note that we do not obtain shorter solutions times or better bounds when we provide CPLEX with our feasible solution obtained using our greedy heuristic. This indicates that the solutions found with the heuristic are of little use to CPLEX.

3.4 Constraint Programming Solution Method and Results

We reformulate MO-CAP as a constraint programming (CP) problem in an attempt to quickly find lower bounds to the problem. We use the Optimization Programming Language (OPL) to formulate the problem using integer variables, where each variable $w_u \in C$ indicates the channel number that unit $u \in U$ is assigned, and the domain of each variable is equal to the number of available channels $|C|$. (We originally formulate this problem using binary variables, but find that the CP solver is much less efficient in determining feasibility using binary variables for this particular problem.)

Table 2. MO-CAP results by time step using pairwise and lazy constraints, and a maximum clique constraint. Bold values indicate an improvement over the previous method. "Time" indicates the time at which the displayed solution and optimality gap is obtained, during a total runtime of 9,000 s.

Time Step	Max Clique Size	Number lazy constraints	Highest-order lazy constraint	Sol'n value	Time (s)	Gap	Improvement over heuristic
1	46	45	6	46	552.68	**0%**	9.80%
2	37	4	4	37	273.40	0%	22.92%
3	34	143	7	**35**	4338.21	**0%**	**23.91%**
4	33	85	6	34	3831.19	**2.94%**	27.66%
5	33	10	3	33	1010.00	0%	23.26%
6	35	95	6	36	3128.34	2.78%	29.41%
7	37	13	6	37	266.68	0%	24.49%
8	30	45	5	31	4415.48	**3.23%**	26.19%
9	32	2	4	32	226.37	0%	25.58%
10	34	6	4	34	323.08	0%	30.61%
11	33	42	8	33	856.69	**0%**	26.67%
12	35	30	5	36	1577.96	2.78%	16.28%
13	31	131	6	32	3172.95	3.12%	25.58%
14	30	214	9	31	2702.16	3.23%	27.91%
15	38	105	6	38	1047.00	**0%**	22.45%
16	35	16	5	36	600.91	**2.78%**	23.40%
17	36	89	5	37	1495.15	2.70%	24.49%
18	31	13	4	31	322.90	0%	22.50%
19	30	74	6	30	1653.29	**0%**	25.00%
20	37	33	4	37	1387.28	0%	24.49%
Aver	**34.4**	**59.8**	**5.5**	**34.8**	**1659.09**	**1.18%**	**24.13%**

We add all pairwise constraints to the problem by indicating that two given units u and v are not allowed to be assigned the same channel, for all pairs $(u, v) \in P$. We solve the problem using IBM ILOG CPLEX CP Optimizer [17].

We begin with a small (infeasible) number of available channels $|C|$ (i.e., the domain of each $w_u \in C$), and iteratively increase $|C|$ until the solver either determines that the problem is feasible, or it cannot resolve the problem within 12 h. At each $|C|$, we have a *relaxation* of the original MO-CAP. If a problem is infeasible with the given number of channels, then we have established that the original MO-CAP (with all constraints) is also infeasible. This indicates that at least $|C| + 1$ channels are required, establishing a MO-CAP lower bound. If the lower bound equals the upper bound (obtained using CPLEX), we have obtained an optimal solution.

Table 3. MO-CAP results by time step using constraint programming. "Optimal solution?" indicates whether the obtained value proves the optimality of a solution, and bolded values indicate new lower bounds (i.e., not found in the previous analyses.

Time step	Infeasible	Optimal solution?
1	45	Yes
2	36	Yes
3	33	
4	32	
5	32	Yes
6	34	
7	36	Yes
8	29	
9	31	Yes
10	33	Yes
11	32	Yes
12	34	
13	31	**Yes**
14	29	
15	37	Yes
16	34	
17	36	**Yes**
18	30	Yes
19	29	Yes
20	36	Yes

The results are displayed in Table 3, where "Infeasible" indicates the largest value at which the solver detects infeasibility, i.e., at least one more channel is required for the problem to be feasible. "Optimal solution?" indicates whether the obtained value proves the optimality of a solution, where bolded values indicate new lower bounds (i.e., not found in the previous analyses). While the solver does not find the exact lower bound at each time step, it does establish two new exact lower bounds (for the problems associated with time steps 13 and 17). When infeasibility is detected by the solver, it is detected extremely quickly (less than a tenth of a second in each case).

On the other hand, the constraint programming procedure was not capable of proving optimality. In general, one can quickly determine that one needs at least k channels (because one can establish that $k-1$ channels are not feasible). If the CP solver cannot establish whether k channels are infeasible within a few seconds, one is likely to find that the solver will not establish the satisfiability of these constraints within 12 h. In order to improve the solver's capabilities to prove optimality, we try adding *symmetry-breaking* constraints (following [34]) but that does not alter the performance result. Next, we try adding all triplet constraints and the known maximum clique constraint (via CP `allDifferent` constraints), as well as adding constraints iteratively, to no avail.

Thus, we conclude that the CP approach is very efficient at finding infeasibilities (and thus establishing lower bounds), but is incapable of finding feasible solutions close to or at the actual lower bound. We conclude that a very good approach to obtaining optimal or near-optimal solutions to the problem is to integrate CP and IP in a complementary fashion ([16]), where IP is used to search for good solutions and establish upper bounds, and CP is used to quickly tighten lower bounds. Similar techniques have been used for large-scale spectrum auctions [20] and scheduling [30,37].

4 Minimum-Cost Channel Assignment Problem over Time (MC-CAP-T)

4.1 Problem Formulation

Given the number of channels needed at a moment in time (established using MO-CAP), a spectrum manager may now wish to reduce the total number of times a radio must change channels. Excessive channel changes waste the time of radio operators and require coordination and synchronization among potentially many dispersed units, which may be difficult to achieve in battlefield conditions.

The *minimum-cost channel assignment problem over time* (*MC-CAP-T*) aims to minimize the cost incurred by channel changes over time, given the number of channels required at each time step. Let the index $t \in T$ represent each time step, and let $g \in G$ (alias h) be a *group* of units that must be assigned the same channel at a given time step. Groups are obtained at each time step from the MO-CAP. A *naïve* approach would simply assign channel numbers to the groups as they appear in order. In practice, this produces surprisingly bad solutions as group membership (i.e., the units assigned to each group) may change significantly from time step to time step, and thus an excessively large cost is incurred if one simply dictates that group 1 is always assigned channel 1, etc. We instead use a *decomposition* approach that takes the solutions from each MO-CAP time step and minimizes the "distance" (i.e., number of channel changes) from one time step to the next. We obtain globally-optimal solutions to this problem in polynomial time using our decomposition approach.

We wish to associate each group g at time t to a group h at time $t+1$ at least cost. Let the binary variable Y_{gh}^t indicate if group g at t is associated with group h at $t+1$, and let $(g,h,t) \in A$ be the arcs representing the set of possible associations between g and h. One could simplify this formulation further by dropping the $t \in T$ index, but we retain the notation to aid in describing our solution approach. At each time step, each group g at t must be assigned a group h at $t+1$, and vice versa, which is enforced via the assignment constraints:

$$\sum_{h \in G} Y_{gh}^t = 1 \quad \forall\,(g,\cdot,t) \in A \tag{17}$$

$$\sum_{g \in G} Y_{gh}^t = 1 \quad \forall\,(\cdot,h,t) \in A. \tag{18}$$

The cost of associating a group g at time t to a group h at time $t+1$, $cost_{gh}^t$, is a function of the difference in unit membership between g and h. Specifically, if $radios_u$ is the number of radios assigned to unit u,

$$cost_{gh}^t = \sum_{u \in h \setminus g} radios_u \quad \forall\,(g,h,t) \in A. \tag{19}$$

That is, the cost from g to h is the number of radios from units that are in group h but not in group g. This method of calculating costs prevents double-counting when a unit moves from an existing channel to a new channel. Our objective function minimizes the sum total costs of associating each group g at t with group h at $t+1$:

$$\min_{Y} \sum_{(g,h,t) \in A} cost_{gh}^t Y_{gh}^t. \tag{20}$$

Note this cost function assumes all units and radios have the same importance, but that need not be the case: one could associate scalar weights with each radio or to an entire unit to model its relative importance.

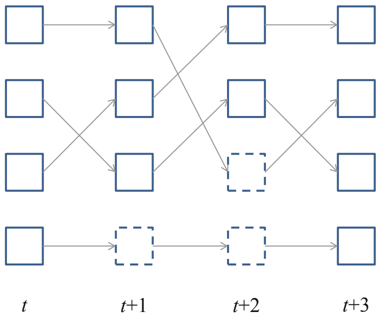

Fig. 2. Example of the association of groups (blue boxes) at each time step. Virtual groups (comprising no units) are represented by dashed boxes. (Color figure online)

4.2 Decomposition Solution Method and Results

To solve the MC-CAP-T, we use a decomposition approach based on the key insight that the actual channel number (or color, or any other label) is arbitrary. We also observe that the cost of changing assignment of a group from t to $t+1$ depends only on the unit membership of each group at t and $t+1$; i.e., the costs can be decomposed by time step. These properties allow us to decompose the problem by time step while maintaining global convergence.

Figure 2 provides a visual representation of the process of associating groups at each time step, where for each time step the column of squares on the left represents groups g and on the right groups h. The number of groups (and their unit membership) is determined by the solutions from the MO-CAP, so some time steps may have more or fewer groups than others. For those time steps with fewer groups than the maximum, we create *virtual groups* (indicated in Fig. 2 by dashed boxes), representing a placeholder group with no assigned units. In this sense, a group represents both a collection of units to be assigned the same channel, and a placeholder for the channel itself, i.e., $|G|$ is equal to the maximum number of available channels across all time steps in the scenario.

At each time step, each group g must be associated with a group h, indicated by gray lines between groups in Fig. 2. When a real group g (i.e., comprising units) is associated with a virtual group h, no cost is incurred because the units in g are assigned to other groups (not in h) at $t+1$. When a virtual group g is associated with a real group at h, the cost equals the number of radios in h, since, according to (18), each unit in h was previously assigned a different group.

We implement our solution in Python. We first calculate all of the $cost_{gh}^{t}$ values for each possible $(g,h,t) \in A$, and then solve a classic integer *assignment problem* at each time step using a variation of the Hungarian (or Munkres) algorithm [21], which solves to optimality in $\mathcal{O}\left(n^3\right)$ time. Global convergence is maintained because at each time step, the cost of channel changes depends only on the assignments at t and $t+1$. The actual assigned channel (i.e., its number) is arbitrary, since all channels provide the same performance and each group must have a channel. Thus this formulation exhibits *optimal substructure* that allows us to efficiently solve each time step to optimality and then combine our results to solve the entire problem to optimality.

Note that in this approach, there is no variable or index representing a particular channel; the association Y_{gh}^{t} implies one. After solving the problem, the *paths* created by associating each g with an h at the next time step represent discrete channels. By assigning a channel number to each of these paths (i.e., the gray lines in Fig. 2), we effectively solve the MC-CAP-T. The following pseudocode describes our algorithm for solving the problem:

Algorithm 1. ***MC-CAP-T***

```
Input: MO-CAP solutions at each time step
Output: $X_u^{ct}, \forall u \in U, c \in C, t \in T$ (unit channel assignments for all time steps)

begin
    Calculate $cost_{gt}^{t}, \forall\,(g, h, t) \in A$
    $channel \leftarrow 1$
    for $g \in G : t = 1$
        $\varGamma_g \leftarrow channel$        // Assign channels to groups during first time step
        $channel \leftarrow channel + 1$
    next;
    for $t = 1, 2, \ldots, t - 1$
        Solve the MC-CAP-T for $t$ using Hungarian / Munkres algorithm
        Store $Y_{gh}^{t}$ values
        for $g, h \in (g, h, t)$
            if $Y_{gh}^{t} = 1$
                $\varGamma_h \leftarrow \varGamma_g$ // Assign channels to groups for time step $t$
            endif;
        next;
    next;
    for $g \in G$
        for $u \in g$
            $X_u^{\varGamma_g^t} \leftarrow 1$        // Assign channels to units
        next;
    next;
end;
```

We solve for each time step in the Marine Corps scenario. The naïve method requires a total of $33,340$ channel changes, whereas our decomposition method (which solves to optimality in less than 53 s) requires $21,915$ channel changes, a reduction of 34%. Figure 3 is a method of visualizing the results of this comparison. For both the naïve and decomposition methods, a row represents a unit, where reddish units are larger (comprising up to 25 radios each) and greenish units are smaller, each column represents a time step, and a blank entry indicates that no channel change is required for that unit at that time step. This visualization provides a qualitative sense of how much better the decomposition method (which provides an exact solution) is at reducing channel changes, especially for larger (and thus more penalizing) units.

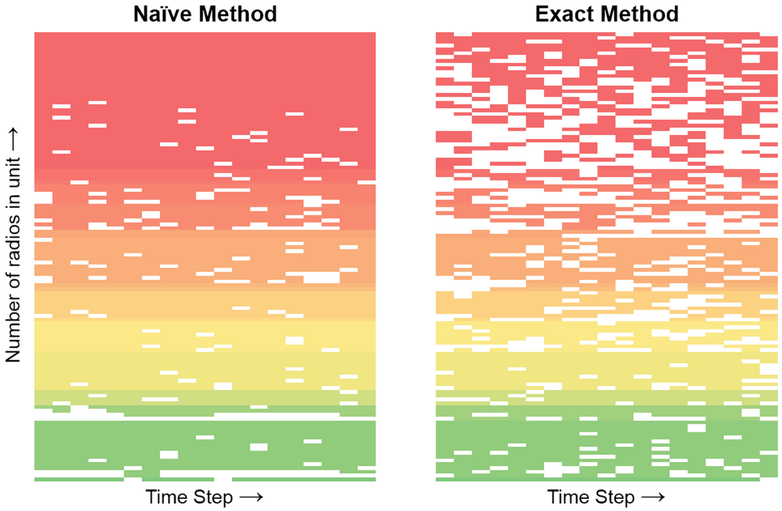

Fig. 3. Results of MC-CAP-T, where each row represents a unit, and each column represents a time step. White indicates that the channel assignment remains the same (i.e., no cost), and color indicates that a different channel is assigned at the next time step. Red indicates larger units (more radios); green indicates smaller units. (Color figure online)

5 Conclusions and Future Research

We present new integer optimization and constraint programming methodologies that solve large, realistic instances of the minimum-order and minimum-cost channel assignment problems to global or near-global optimality in reasonable amounts of time. Our approach can be used to support military spectrum managers who must quickly make spectrum allocation decisions in congested EM environments. Our ongoing and future research explores robustness and resiliency in the presence of an adversary determined to jam portions of the EM spectrum (see, e.g., [10,22,29,39]).

References

1. Aardal, K.I., Van Hoesel, S.P., Koster, A.M., Mannino, C., Sassano, A.: Models and solution techniques for frequency assignment problems. Ann. Oper. Res. **153**(1), 79–129 (2007)
2. Alion Science and Technology Corporation: TIREM RF Modeling (2017). https://www.alionscience.com/terrain-integrated-rough-earth-model-tirem/
3. Analytical Graphics Inc.: Engineering Tools (2018). http://www.agi.com/products/engineering-tools
4. Berry, L.: The potential contribution of optimum frequency assignment to efficient use of the spectrum. In: IEEE International Symposium on Electromagnetic Compatibility, pp. 409–412. IEEE (1990)

5. Bron, C., Kerbosch, J.: Algorithm 457: finding all cliques of an undirected graph. Commun. ACM **16**(9), 575–577 (1973)
6. Cuppini, M.: A genetic algorithm for channel assignment problems. Eur. Trans. Telecommun. **5**(2), 285–294 (1994)
7. Daniels, K., Chandra, K., Liu, S., Widhani, S.: Dynamic channel assignment with cumulative co-channel interference. ACM SIGMOBILE Mob. Comput. Commun. Rev. **8**(4), 4–18 (2004)
8. Department of Defense: Integrated Security Construct-B. Multi-Service Force Deployment, scenario 3 (2013)
9. Dunkin, N., Bater, J., Jeavons, P., Cohen, D.: Towards high order constraint representations for the frequency assignment problem. University of London, Egham, Surrey, UK, Technical report (1998)
10. El-Bardan, R., Brahma, S., Varshney, P.K.: Power control with jammer location uncertainty: a game theoretic perspective. In: 48th Annual Conference on Information Sciences and Systems, pp. 1–6. IEEE (2014)
11. Fischetti, M., Lepschy, C., Minerva, G., Romanin-Jacur, G., Toto, E.: Frequency assignment in mobile radio systems using branch-and-cut techniques. Eur. J. Oper. Res. **123**(2), 241–255 (2000)
12. Github: CAP datasets (2019). https://github.com/nickelpickle1/cap_dataset/
13. Goldstein, P.: Pentagon strikes deal with broadcasters, clearing way for 1755–1780 MHz auction. Fierce Wireless, February 2013. http://www.fiercewireless.com
14. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using NetworkX. In: Proceedings of the 7th Python in Science Conference (SciPy 2008), pp. 11–15. Pasadena, CA USA, August 2008
15. Hart, W.E., Laird, C., Watson, J.P., Woodruff, D.L.: Pyomo-Optimization modeling in Python, vol. 67. Springer Science & Business Media, New York (2012). https://doi.org/10.1007/978-1-4614-3226-5
16. Hooker, J.N., Ottosson, G.: Logic-based Benders decomposition. Math. Program. **96**(1), 33–60 (2003)
17. IBM: IBM CPLEX Optimization Studio (2018). http://www-01.ibm.com/software/commerce/optimization/CPLEX-optimizer/
18. Johannson, F., et al.: mpmath: A Python library for arbitrary-precision floating-point arithmetic (version 0.18) (2013). http://mpmath.org
19. Katzela, I., Naghshineh, M.: Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey. IEEE J. Pers. Commun. **3**(3), 10–31 (1996)
20. Kiddoo, J., et al.: Operations research enables auction to repurpose television spectrum for next-generation wireless technologies. INFORMS J. Appl. Anal. (2018, submitted)
21. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Logistics Q. **2**(1–2), 83–97 (1955)
22. London, J.P.: The new wave of warfare-Battling to dominate the electromagnetic spectrum. J. Electron. Defense (JED) **38**(9), 68–76 (2015)
23. Mannino, C., Sassano, A.: An enumerative algorithm for the frequency assignment problem. Discrete Appl. Math. **129**(1), 155–169 (2003)
24. Margot, F.: Symmetry in integer linear programming. In: Jünger, M., et al. (eds.) 50 Years of Integer Programming 1958–2008, pp. 647–686. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-540-68279-0_17
25. Metzger, B.: Spectrum management technique. In: 38th National ORSA Meeting (1970)

26. Nicholas, P.J.: Optimal spectrum allocation to support tactical mobile ad-hoc networks. Ph.D. thesis, George Mason University (2016)
27. Nicholas, P.J., Hoffman, K.L.: Computational challenges of dynamic channel assignment for military MANET. In: Proceedings of the Military Communications Conference (MILCOM), pp. 1150–1157. IEEE (2015)
28. Nicholas, P.J., Hoffman, K.L.: Optimal channel assignment for military MANET using integer optimization and constraint programming. In: Proceedings of the Military Communications Conference (MILCOM), pp. 1114–1120. IEEE (2016)
29. Nicholas, P.J., Hoffman, K.L.: Analysis of spectrum allocation to support mobile ad-hoc networks in contested environments. In: Proceedings of the Military Communications Conference (MILCOM), pp. 145–150. IEEE (2018)
30. O'Neil, R.J., Hoffman, K.: Integer models for the asymmetric traveling salesman problem with pickup and delivery (2018). optimization-online.org
31. Ostrowski, J., Linderoth, J., Rossi, F., Smriglio, S.: Orbital branching. Math. Program. **126**(1), 147–178 (2011)
32. Palpant, M., Oliva, C., Artigues, C., Michelon, P., Didi Biha, M.: Models and methods for frequency assignment with cumulative interference constraints. Int. Trans. Oper. Res. **15**(3), 307–324 (2008)
33. Poisel, R.: Modern Communications Jamming: Principles and Techniques. Artech House (2011)
34. Ramani, A., Aloul, F.A., Markov, I.L., Sakallah, K.A.: Breaking instance-independent symmetries in exact graph coloring. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, vol. 1, pp. 324–329. IEEE (2004)
35. Selyukh, A.: In switch, U.S. military offers to share airwaves with industry, Thomson Reuters (2013)
36. Tomita, E., Tanaka, A., Takahashi, H.: The worst-case time complexity for generating all maximal cliques and computational experiments. Theor. Comput. Sci. **363**(1), 28–42 (2006)
37. Trick, M.: Sports scheduling. In: Van Hentenryck, P., Milano, M. (eds.) Hybrid Optimization: The Ten Years of CPAIOR, pp. 489–508. Springer Science & Business Media, New York (2010)
38. Wang, S.W., Rappaport, S.S.: Signal-to-interference calculations for balanced channel assignment patterns in cellular communications systems. IEEE Trans. Commun. **37**(10), 1077–1087 (1989)
39. Wu, Y., Wang, B., Liu, K.R., Clancy, T.C.: Anti-jamming games in multi-channel cognitive radio networks. IEEE J. Sel. Areas Commun. **30**(1), 4–15 (2012)