Space Allocation Optimization at NASA Langley Research Center

Rex Kincaid*

*Department of Mathematics, College of William & Mary
Williamsburg, VA, 23187-8795, USA
rrkinc@wm.edu

1 Introduction

NASA has undergone several reorganizations and workforce realignment exercises. During NASA Langley Research Center's (LaRC) effort to re-organize and simultaneously re-allocate personnel in 2004, the need for an optimization tool for space utilization and allocation arose. A bin packing approach to the problem was implemented as a first cut at building an optimization tool and has since been replaced by a local improving search. Although the resulting model did not fully capture all aspects of the problem, several significant visualization tools were developed. NASA-LaRC has roughly 6,200 rooms and 300 buildings housing more than 3,500 individuals and 1,600 research labs. A related research problem is office space allocation in a university setting. A research group at the University of Nottingham (see references [1]—[7]) has a series of papers on this topic.

2 Problem Definitions

A complete personnel move consists of two parts: moving an individual *from* one building and office location *to* a new building and office location. There are a variety of allocation and scheduling problems that arise. Here, we focus on two. First is the weekly space allocation problem (WSAP). Even without a major reorganization NASA-LaRC makes many personnel moves during a typical year. Typical moves include retirements, employee separation for other jobs, new hires, and project reassignments. In section 3, we will see that there are approximately 20 moves per month (range 3 to 50). Closing buildings and moving research labs are not considered in this problem formulation. Thus, the resulting optimization model is smaller in scale. The goal is to have an algorithm that generates high quality solutions in real time. A second problem is the large-scale reorganization space allocation problem

(LRSAP). The most recent large-scale reorganization occurred in 2004. All research labs and potential building closures are included in this model. The visualization tools needed for both of these problems are nearly identical. The run-time goal for LRSAP is to find good solutions in less than a few hours.

3 Move Data Analysis

The NASA-LaRC move tool is able to provide historical data on all personnel moves from July 2004 through the present. This period covers moves that were implemented during a major reorganization as well as the more typical move pattern during a stable organizational structure. One of the motivations for taking an extended look at the personnel move data is to understand how solutions to LRSAP might survive the changes imposed by the natural ebb and flow of personnel moves. A second reason for analyzing the move data is to determine the number and type of moves per year (per month) during a stable organizational period.

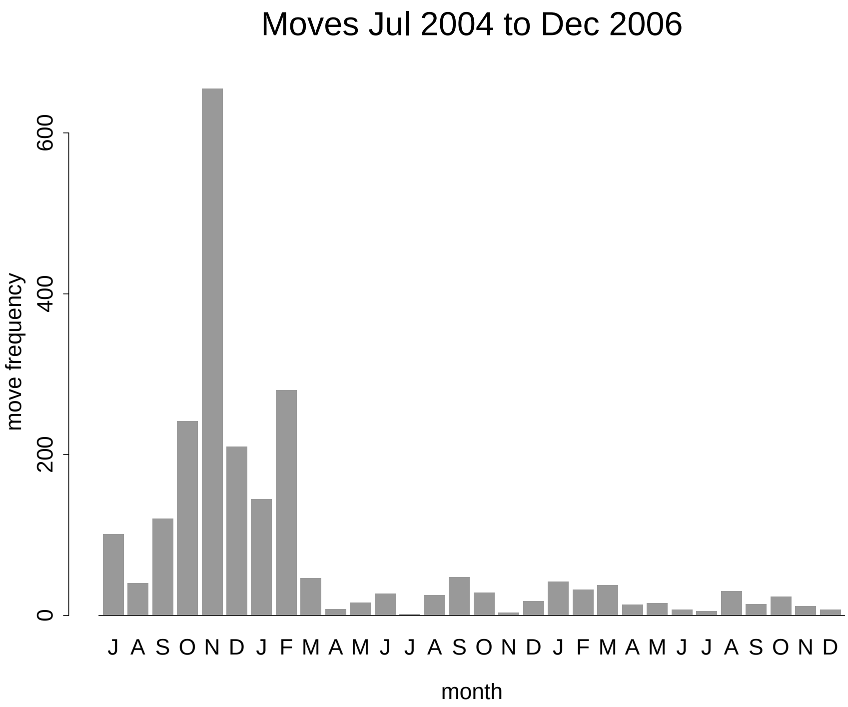
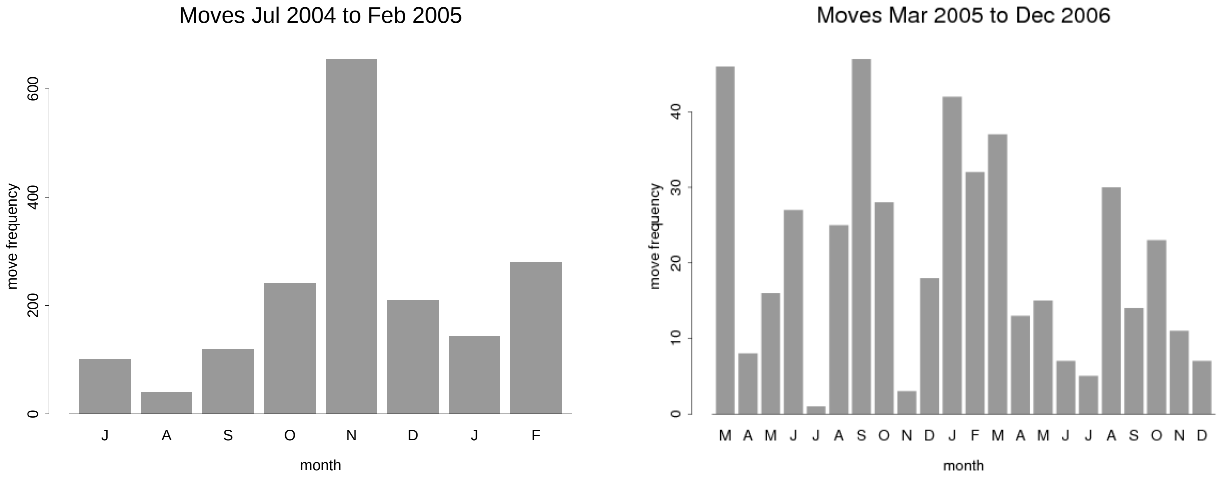


Figure 1: Personnel move data: 30 month time period

Figure 1 displays data gathered from the NASA move tool over a 30 month period—July 2004 through December 2006, Clearly the move distribution partitions into two distinct types. The first, a large-scale reorganization, is from July 2004 through February 2005. The second, a normal operation phase, is from March 2005 through December 2006. Figures 2a and 2b provide separate bar charts for these two time periods. (Please note the difference in



(a) July 2004 - February 2005

(b) March 2005 - December 2006

Figure 2: Personnel move data: two distinct time periods

the y-axis scales.) Time period 1: 7.04-2.05 is comprised of $1,791$ moves while time period 2: 3.05-12.06 has only 455 moves.

Figure 3 catalogs the number of moves (From and To) for each building over the entire time period (30 months). In order to smooth out the scale on the x-axis building building numbers 647,648, 1000 and 1001 were mapped to 1107, 1108, 1310 and 1311 respectively. Qualitatively the plots in Figures 3a and 3b are quite similar leading us to conclude that their are no significant qualitative differences in the number of moves into and out of each building. With regard to maintaining the structure of a solution found for either WSAP or LRSAP the net change in a building over any time period appears to be more critical than the total number of moves from and to each building. For the first time period 332 (18.5%) of the 1791 total moves available remain in the same building. For the second time period (normal operations) there are 455 total moves. Of these only 13 (2.8%) remain in the same building.

A natural alternative measure is to count the net number of moves *from* and *to* each building. For time period 2 the sum of the net number of personnel moves per building is 326 (out of the 455 available moves). That is, fully 28% of the moves do not change the building capacity during the 22 month time period. Hence, under normal operating conditions only 326 moves are likely to disrupt any space allocation solution. If we assume a work force size of $3,500$ then 250 moves per year $(12 \cdot 455/22))$ yields a 7% personnel move rate. However, if we only consider moves that change the building capacity then the move rate is closer to 5% per year. Given the relatively low disruption rate (5%) it seems likely that solutions to WSAP and LRSAP will maintain their integrity.

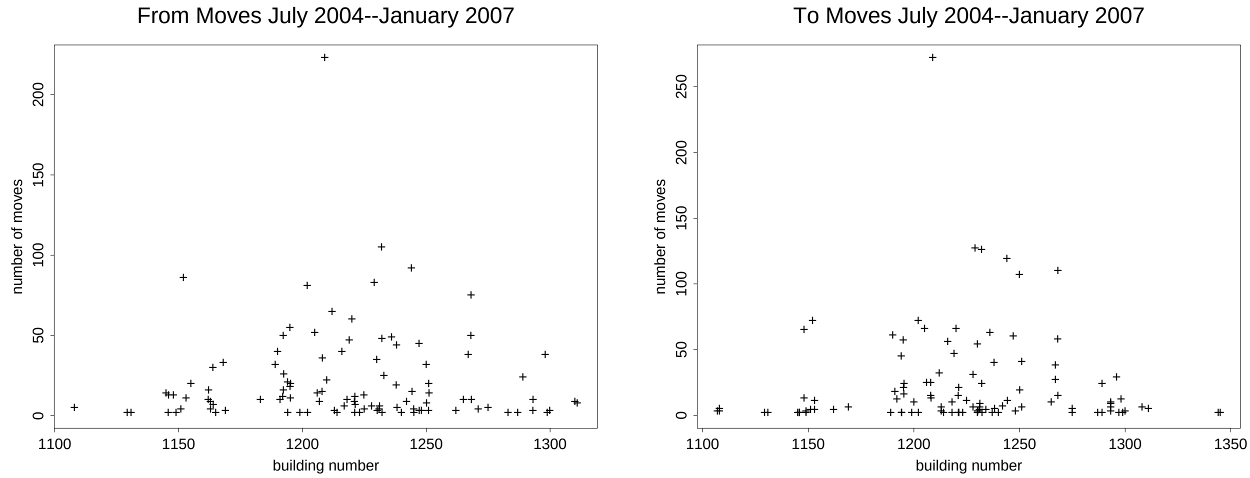


Figure 3: Personnel move data: 30 month time period

4 Review of Current Literature for Space Allocation

The class of Space Allocation or Capacity Allocation problems are those in which the amount of space (area or volume) or capacity available is to be distributed among a set of items, satisfying specific requirements and constraints. Examples of this class of problems are: bin packing, knapsack problem, and space planning.

The *Automated Scheduling and Planning Group (ASAP Group)* in the Department of Computer Science at the University of Nottingham, UK has maintained a focused effort to address the space allocation problem in the context of academic institutions. Their efforts began in 1998 and continue to the present. Although they suggest that typical real instances range from 1600 rooms in 30 buildings to 20,000 rooms in 600 buildings, the test cases presented in their research are on the order of 150 rooms. Most real instances of their Office Space Allocation problem can be classified as one of the following types:

Reorganization of the existing allocation is the rearrangement of the current space distribution among the entities. It is performed to improve the existing solution under the existing conditions or to modify the allocation due to changes in the conditions (requirements, constraints, number of entities to be allocated, number of areas of space) of the problem. This problem is nearly the same as WSAP.

Construction of a complete solution is the generation of a new solution from scratch to distribute all available areas of space among all entities under the given conditions. Note the similarity to LRSAP.

When reorganizing an existing allocation, a constraint to minimize the amount of disruption caused, (limit the number of entities moved) may be considered. Alternatively, a budget constraint for the total cost of all moves may be imposed to achieve the same purpose. When reorganizing an allocation, the amount of disruption permitted establishes a balance between the quality of the allocation and the difficulty in achieving it.

4.1 Problem Statement

The typical office space allocation problem consists of a number of *hard* and *soft* constraints. The constraints restrict the relative position and grouping of the entities. *Proximity* constraints exist when an entity should be close to certain rooms or to other entities. For example, a department secretary near (or adjacent to) a department head. *Grouping* constraints require that a group of entities should be allocated close to each other. For example, mathematics department members should be located together. *Sharing* constraints indicate that certain entities should *not* share a room. Here, an example is a department chair that is not allowed to share space with anyone else. *Space usage* constraints provide targets for the number of square feet to be allocated to each entity. Penalties for too much or too little space allocated are imposed. *Location* constraints force entities to reside in a specific room location. An example is a computer lab network technician who is pinned to the lab he services. The goal is to allocate space so that utilization is maximized, all hard constraints are satisfied and as many soft constraints are met as possible.

Let n be the number of entities with corresponding sizes s_i for $i = 1, \ldots, n$. Examples of entities are department chairs, faculty members, graduate students and office administrators. Let m be the number of rooms with corresponding capacities c_i for $i = 1, \ldots, m$. Let x_{ij} denote the binary decision variables. $x_{ij} = 1$ if entity j is assigned to room i and 0 otherwise. Let X denote the matrix of x_{ij} values. The optimization problem is

$$
\begin{array}{rc}
\text{min} & f_1(X) + f_2(X) \\
\text{subject to} & \\
 & \sum_i x_{ij} = 1 \text{ for } j = 1, \ldots, n \\
 & \text{hard constraints} \\
 & \text{soft constraints}
\end{array}
$$

$f_1(x_{ij})$ measures the deviation (positive or negative) of the target, s_j from the actual space assigned, c_i. $f_2(X)$ records the penalties (if any) for violation of any of the soft constraints. All hard constraints must be satisfied or the solution is deemed infeasible.

4.2 Search Technique History

The ASAP Group demonstrated early on (1998) that bin packing algorithms failed to yield solutions of interest. Bin packing optimizes the f_1 objective (f_2 is not present). The ASAP group showed that solutions that are high quality with respect to f_1 are of low quality with respect to f_2. Moving away from the myopic bin packing approach the ASAP Group next examined hill climbing strategies (local search), simulated annealing, and genetic algorithms.

In [2] and [4] the authors found that GAs (population based) were outperformed by hill climbing and simulated annealing. Not until [7] were they able to find a way to make significant use of a population based heuristic approach (more on this paper later). In [3] the ASAP Group tested a more comprehensive (and complicated) metaheuristic that combined previously successful features of both hill climbing and simulated annealing from [4]. Nothing of note here except that the bar graphs plotting the contribution of f_1 and f_2 to the total performance metric indicates that the f_2 objective's contribution dominates by at least 2 to 1 margin and sometimes as much as 10 to 1.

The next paper (chronologically) is [5]. The point of interest to me in this paper is not the heuristic that is developed and tested, but the focus on the office space allocation problem as a bi-objective optimization model. The author's show that minimizing either f_1 or f_2 alone results in increasingly poor performance for the non-optimized metric. If f_1 is minimized as a single objective the associated contribution of f_2 increases as f_1 decreases. The effect is less pronounced when f_2 is minimized as a single objective. Next the authors compared two ways for computing the performance measure. Method one adds the two objectives, $f_1 + f_2$. Method two maintains a vector of objective values, (f_1, f_2). In method two improving solutions are those that dominate (in a component-wise vector sense). The result of the comparison was that method one performed the best.

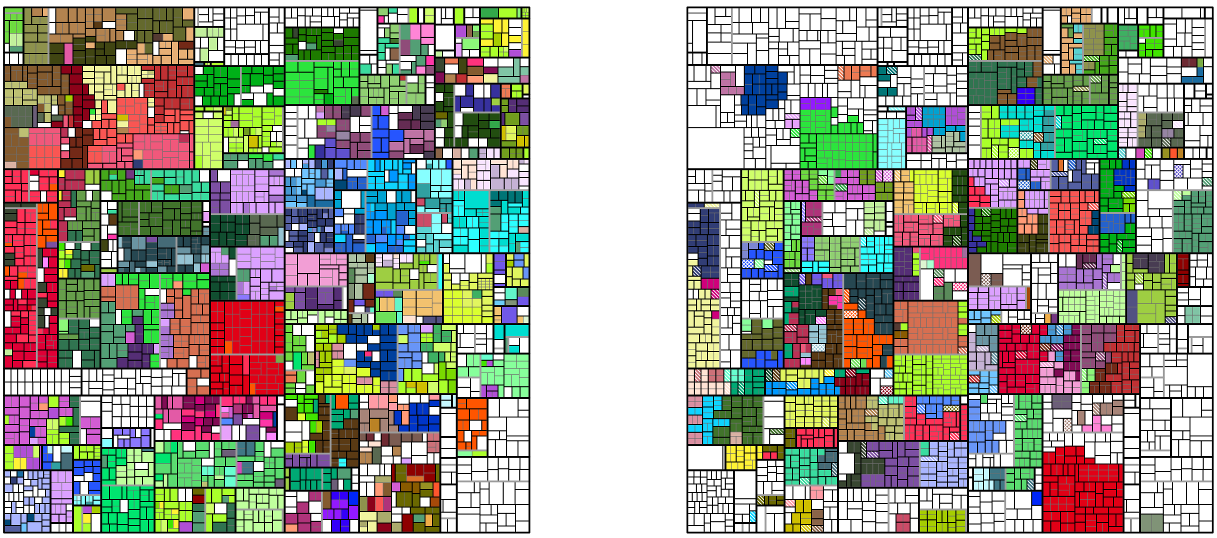
The last paper examined from the ASAP Group is quite recent. [7] is scheduled to appear in a high quality journal in 2007. I am encouraged that the paper is appearing in a well respected journal since all of the previous work appeared solely in conference proceedings. The authors test local search (hill climbing), simulated annealing, and tabu search heuristics as well as extend each of these single solution heuristics to a population based variant (asynchronous cooperation). Given their earlier failure with genetic algorithms (local search won out), this is an inventive way to include population based information. In addition, the non-standard scheme yields higher quality solutions. The upshot of their computational experiments is that population based variants of tabu search and their heuristic in [3] were the winners. The description of how the single solution heuristics are extended via asynchronous cooperation will not be included here. However, a few comments of interest concerning details of their single solution search algorithms is warranted.

Three types of moves are allowed in attempting to improve a solution. The first, a *swap* move, is as expected. Assume entity A is assigned room 12 and entity B is assigned room 7. A swap move puts B in room 12 and A in room 7. The second move is called *relocate* and simply changes the room assignment of a single entity. In our previous example, *relocate* entity A from room 12 to room 17. (This assumes the space in room 17 has not yet been assigned.) Last, an *interchange* move exchanges a block of assigned spaces between two rooms. Basically a group *swap* move.

The implementation of tabu search is relatively simple. How long a move remains tabu is static. (There are some versions of tabu search that dynamically alter the tabu tenure of a move based upon search terrain characteristics.) A simple *aspiration* criterion is employed. That is, a move that is tabu may be accepted if it leads to the best solution seen thus far. (The goal of making moves tabu is to avoid cycling back to previously observed solutions and/or retracing portions of the search trajectory. Clearly, a tabu move leading to the best solution ever seen is not repeating any portion of the search trajectory.) Two pieces of information during the search are stored in matrices, M_T and M_A. Entries $m^T_{j,i}$ and $m^A_{j,i}$ corresponds to the allocation of entity j to room i. For example, if on iteration 93 entity 6 is moved from room 2 to room 4 then $m^T_{6,4} = 93 + tabu\_tenure$. If, in addition, the move yields an improved solution then $m^A_{6,4}$ is incremented by one. (Initially M_T and M_A contain only zeroes.) The M_A matrix is an example of long term memory and can be used when the search stalls to restart the search.

5 Visualization Tool and Computational Experience

Figure 4 provides an example of one of the NASA visualization tools. In it every room in every building is shown. The colors indicate research group affiliation. Hence, keeping the same colors in proximity is a good thing. In addition, white space indicates that the space is unassigned. It is easy to see the qualitative improvements in the allocation of space and the proximity of colors. We note that although the optimization procedure makes no explicit attempt to close buildings (no cost saving metric for this is included) the local optimum has several buildings that are empty. There is a significant cost savings if a whole building can be vacated and closed.



Current Configuration Local Optimum

Figure 4: Visualization Tool for Space Allocation Solutions.

In addition to the visualization of solutions provided in Figure 4, it is also possible to compare the raw cost data across various solutions. There are four metrics that are summed which yield the final cost for each solution. The first metric is the cost of the actual moves needed to transform a starting space allocation to the locally optimal one. The second metric measures the office space constraint violations. There are penalties for missing the targets from above or from below. Metric three is the sum of all the individual synergies due to group assignments across the center. Lastly, the sum of all the functional area synergies comprise metric four. Table 1 below summarizes the components of several different locally optimal solutions. The optimization procedure takes about 4 to 6 minutes to find each of these solutions. The first six rows of the table contains local optima that were found by starting with a random feasible solution while the last row is the local optimum that is found when starting with the current NASA space allocation. Notice that the starting with the current configuration results in dramatic improvements in metric 1 (move cost), metric 2 (office space constraint violations), and metric 4 (functional area synergy). An obvious explanation for the observed difference in metric 1 is that the current configuration is much closer to a local optima than are randomly generated solutions. Similarly, it is likely that metric 2 is smaller since the current configuration has made more reasonable allocations of the office spaces than would a random configuration. I know of no obvious reason why the personal synergy metric 3 is on par with the random starts but the functional area metric 4 is not.

Table 1: Cost metrics for several local optima.

Start Soln	**Metric 1**	**Metric 2**	**Metric 3**	**Metric 4**	**Total Cost**
Random 1	219,000	465,505	13,458,027	2,258,440	16,400,972
Random 2	208,500	485,851	13,054,669	1,933,836	15,682,855
Random 3	215,800	475,588	13,372,742	1,961,452	16,035,582
Random 4	216,800	488,121	13,173,170	1,912,015	15,790,107
Random 5	214,400	455,853	13,808,311	2,142,936	16,621,501
Random 6	211,900	451,417	13,287,075	2,251,814	16,202,307
Current	146,600	274,460	13,292,463	1,386,828	15,100,351

6 Discussion and Recommendation

There is a striking similarity in the understanding of the space allocation problem uncovered by the ASAP and NASA research groups. Both groups first tried bin packing approaches and both abandoned this single objective scheme in favor of a multiple objective model. Both groups found that typical population based heuristics fail to be effective. The ASAP group has a well documented account of this failure in [4].

The method by which the competing objectives are included, however, are quite different. Both groups use violations of square footage targets for each entity to which space is allocated as objective one (f_1 in section 4.1). The ASAP group chose a more traditional optimization approach by adding a second objective, f_2, that penalizes violations of a variety of *soft* constraints (see section 4.1). To understand the NASA approach it is convenient to think of each entity that is to be assigned a space as a particle. Certain particles have a greater affinity or attraction for each other. That is, two highly attracted particles' *synergy* is maximized by sharing adjacent office spaces. Maximizing the collective synergy of all particles is the goal of this objective.

ASAP identified these new objectives as having the greatest importance (see section 4.2) in [3]. The NASA group has also independently verified that this is true for the synergy objective as well. The ASAP group's efforts in [5] nicely illustrate the conflicting role of the two objectives f_1 and f_2 and that procedures that attempt to optimize a weighted sum of the two objectives dominated attempts identify efficient or pareto-optimal solutions. The NASA group has also taken the weighted sum approach for their optimization model.

The current optimization approach taken by the NASA group has two phases. First, a greedy neighborhood interchange heuristic identifies a local optimum. (The starting solution is either chosen at random or is the current space configuration.) Next, a merge procedure is implemented which seeks to inject portions of the local optimum into a randomly generated solution. The merge procedure has been shown to generate solutions that are approximately 5-10 percent better than any local optima.

There are several refinements and/or additions I would suggest for the current NASA optimization procedure.

1. Add simple tabu search features. A tabu tenure for each move and an aspiration criteria as described in section 4.2. The addition of a group (branch) move would be useful as well.

2. Devise a constructive heuristic to build feasible solutions. Once done, the M_A matrix (section 4.2) can be used in the constructive phase.

3. Construct a path re-linking search to uncover high quality solutions between distinct local optima.

4. Utilize features of the merge procedure to create a population of high quality solutions and incorporate features of asynchronous cooperation outlined in [7] or as in scatter search.

5. Develop a method for building the underlying group interaction network. The current approach simply connects everyone to each other. It is more typical for these type of networks to follow a power law.

In addition to the above refinements there are other questions of interest. For example, is it possible to build robustness into the performance metrics? A robust solution would be able to absorb the monthly personnel moves without a significant degradation in the four metrics. Perhaps a simulation model driven by last year's set of moves (or sampled from an appropriate distribution) to see how different solutions degrade over a year's time would be beneficial. The same simulation model could also be used to evaluate the effectiveness of different move schedules during a major reorganization. A second issue of interest is how to get center employees to buy into the solutions generated by the optimization procedure. Would an online tool allowing anyone to try out their pet solution be useful? Would it be better to allocate space at the branch level and then let the branches allocate their total space however they want? Space allocation optimization is a problem ripe for further development. As we heard during the panel discussion at the recent ESRI users conference, every branch of the military is faced with this problem and has few tools to address it. Further development of portable optimization and visualization software will be of great interest to nearly every government/military agencies.

References

[1] Burke, E.K., C. Beyrouthy, J.D. Landa-Silva, B. McCollum, P. McMullan (2004): "SpaceMAP Applying Meta-Heuristics to Real World Space Allocation Problems in Academic Institutions". In: *Proceedings of the 2004 International Conference on the Practice and Theory of Automated Timetabling (PATAT 2004)*. Pittsburgh USA, 441–456.

[2] Burke, E.K. and D.B. Varley (1998). "Automating Space Allocation in Higher Education," In: *Proceedings of the 2nd Asia Pacific Conference on Simulated Evolution and Learning*, Camberra, Australia, pp. 66-73.

[3] Burke, E.K., P. Cowling and J.D. Landa-Silva (2001a) "Hybrid Population-based Metaheuristics Approaches for the Space Allocation Problem," *Proceedings of the 2001 Congress on Evolutionary Computation*, pp. 232-239.

[4] Burke, E.K., P. Cowling, J.D. Landa-Silva, and B. McMullan (2001b) "Three Methods to Automate the Space Allocation Process in UK Universities," *The Practice and Theory of Automated Timetabling III: Selected Papers from the 3rd International Conference on the Practice and Theory of Automated Timetabling (PATAT 2001)*. Lecture Notes in Computer Science **2079**, pp. 254-273, Springer, New York.

[5] Burke, E.K., P. Cowling, J.D. Landa-Silva and S. Petrovic (2001c) "Combining Hybrid Metaheuristics and Populations for the Multiobjective Optimisation of Space Allocation Problems," *Proceedings of the 2001 Genetic and Evolutionary Computation Conference (GECC 2001)*.

[6] Landa-Silva, J.D. (2003) Metaheuristic and Multiobjective Approaches for Space Allocation, Ph.D. Thesis, Automated Scheduling and Planning Group, Department of Computer Science, University of Nottingham, UK.

[7] Landa-Silva, J. Dario and Edmund K. Burke (2007) "Asynchronous Cooperative Local Search for the Office Space Allocation Problem". *INFORMS Journal on Computing*, to appear.