Performance Evaluation of Tabu Search and Ant Colony Optimization Algorithm in Solving Office Space Allocation Problem (OSA)

Agboola O.M., Abiodun, E. & Ayeni, J. K
Kwara State Polytechnic
Ilorin, Nigeria
E-mail: oladiranagb@gmail.com; etabiodun1492@gmail.com; kennybetty2006@gmail.com

ABSTRACT

Space is an area that is occupied by or designed for an individual or thing, that is considered one of the most valuable assets in a standard organization. while Office Space Allocation (OSA) problem is one of the classes of complex combinatorial optimization problem that involves the distribution of a set of entities to a set of room while satisfying a set of requirements and constraints (hard and soft). This work aimed at discovering an allocation that optimizes the use of space and time as much as possible. This paper presents an evaluation of tabu search and ant colony optimization algorithm in solving the OSA problem. After series of iteration, it was discovered that tabu search algorithm smart performs the ant colony optimization algorithm in terms of average execution time of 0.62842529 as against Ants' colony 1.63200581(10 iterations). The benchmark instances proposed by the University Teaching Commission for tertiary institution were employed in the evaluation of the office space allocation problem. The research was implemented using visual C# to develop a console application as the frontend while Microsoft SQL Server was use for the backend.

Keywords: Office Space Allocation, Constraint, Ant Colony, Tabu Search, Optimization.

iSTEAMS Proceedings Reference Format
Agboola O.M., Abiodun, E. & Ayeni, J. K. (2019): Performance Evaluation of Tabu Search and Ant Colony Optimization Algorithm in Solving Office Space Allocation Problem (OSA). Proceedings of the 19th iSTEAMS Multidisciplinary Conference, The Federal Polytechnic, Offa, Kwara State, Nigeria. 7th – 9th August, 2019. Pp 253-260. www.isteams.net/offa2019 - DOI Affix - https://doi.org/ 10.22624/AIMS/iSTEAMS-2019/V19N2P26

1. INTRODUCTION

Space is an area that is occupied by or designed for an individual or thing that is considered one of the most valuable assets in a standard organization. The Office Space Allocation (OSA) seeks to use space resources efficiently to minimize or eradicate the misuse of space. The space allocation assignment may include constraints and goals preferred an organization. The distribution of office space in any organization is generally a problematical issue, which regularly requests a significant measure of time to perform physically. This distribution's result affects the life of anyone who uses the office. In academic institutions, the distribution of the available room space to staff is a process that needs to be carried out on a regular basis because of the continuous changes that occur in this environment. For example, when new employee are given appointment, people leave the institution or move to another department/faculty, new lecture rooms or labs are required, offices for new staff or research students should be available, certain rooms are unavailable for various reasons.

Space Allocation Problem

Space distribution is one of the complicated problems of combinatorial optimization that has some similarities with classical knapsack issues (Martello & Toth, 1990) and is also linked to scheduling issues. In the traditional knapsack problem, a set of objects of certain sizes must be accommodated in a set of capacity containers so that the capacities available are used as effectively as possible, but there are generally no extra constraints (Wren, 1996). The issue in academic scheduling is to accommodate a number of events in the set of time slots accessible to satisfy extra constraints. The variety of techniques used to address combinatorial optimization issues can be categorized into two particular groups: exact methods and approximate (heuristic) methods Papadimitriou and Steiglitzz (1999). Exact techniques try to provide the accurate result to a given problem in order to ensure optimal performance, but their execution time generally involves too much computing time. Heuristic techniques try to discover (not necessarily ideal) high-quality alternatives within reasonable computational times for practical use.

The manual process for allocating space

There is a centralized office in most academic institution that controls the allocation of space and assignment of space areas to faculties, colleges, departments, among others. Space officers and administrators are responsible for building allocation at distinct levels. The space needed for each entity, the room space available, the constraints to be met (hard constraints), those that are desirable to be meet (soft constraints) and additional requirements are determined. Information on accessible space fields (size, place, closeness,) is acquired with the help of office spaces and room databases. Entities are assigned according to the particular development of the rooms in order of relevance. Every time an entity is assigned, the satisfaction of space requirements and constraints is reviewed. A solution assessment includes various requirements, and in some instances these criteria may come from various decision-makers. Due to the nature of the manual method, it is prevalent for a final solution to be obtained for weeks or months.
Considering the allocation complexity, there is need to come up with a computerized solution, in order to ease the allocation process. To this extent, this work intend to implement the computerized Office Space Allocation problem using metaheuristic Algorithms(Ant Colony and Tabu search). The results of the two algorithms will be compare also to know which algorithm gives the best result in terms of time used in the allocation.

2. REVIEW OF RELATED PAST WORKS

Asaju, Ikechi and Peter (2017) presented a research paper titled optimization of office space allocation using artificial bee colony, the paper explore an Artificial Bee Colony Algorithm (ABC) for the Office-space allocation (OFA) problem using the datasets published by University of Nottingham and University of Wolverhampton respectively. The initial feasible solutions are generated by the proposed techniques using Peckish heuristic strategies. ABC as an iterative improvement algorithm optimizes the food sources with the aid of three operators: employed, onlooker and scout bees. The best food source generated at each cycle is memorized. This search process is repeated until a maximum cycle number (MCN) is achieved. Experimental results produced by the proposed technique are compared with those generated by other existing techniques, which shows the promising performance of the techniques. Interestingly, the proposed ABC produced two new results one best result and comparable results on the remaining instance from the dataset of the University of Nottingham. They further recommend that an improvement could be tailored towards enhancing the exploitation capability of the proposed ABC when employed to solve the OSA. Therefore, they recommend that the future work will focus on the enhancement of this technique.

Ulker and Dario (2015) Proposed a 0/1 integer programming model to address the problem of office space allocation (OSA), which refers to allocating space to a set of entities (people, machines, roles) with the aim of optimizing space utilization while meeting a set of additional requirements. They further discussed that conditions can be modeled as constraints (hard constraints) or goals (soft constraints) in the suggested strategy. They conducted that some experiments on benchmark instances and note that setting certain constraints as hard or soft affects this issue of combinatorial optimization on the computational problem significantly. Their findings show that setting all kinds of constraints as hard makes the problem unsolvable. It also seems particularly difficult to set the fulfillment of the same room constraint as soft.

Future work is to modify the IP model in order to improve computational time and memory consumption, develop a general IP model and develop hybrids between the IP model and heuristics. For the benchmark instances considered, they compare the best results from their experiments with the best results reported in the literature. They have been able to significantly enhance on the past outcomes.

Asaju, Ikechi and Peter (2019) used an adaptation of Late Acceptance of Hill Climbing LAHC algorithm to tackle the OSA problem in which three neighborhood structures are embedded with the operators of the LAHC algorithm in order to explore the solution space of the OSA efficiently. The benchmark instances proposed by the University of Nottingham and University of Wolverhampton data sets are employed in the evaluation of the proposed algorithm. The LAHC algorithm is able to produce one new result, two best results and competitive results when compared with the state-of-the-art methods.

Their paper further investigates a Late Acceptance Hill Climbing (LAHC), a recently proposed one-point meta-heuristic algorithm for solving Office-Space Allocation Problem. The LAHC algorithm is a variant of hill climbing optimizer that utilized an advanced acceptance criterion (Late acceptance strategy) in order to prevent the algorithm from getting stuck in a local minimum. The design of the experiment is intentionally made to test effects of the different combinations of these neighborhood searches on the performance of the proposed LAHC. The computational results proved that the LAHC incorporated with the three neighborhood searches is an effective technique for the OFA.

Pereira, Kevin and Rex (2015) presented a research work titled office space allocation optimization" In large organizations, the allocation of buildings and office space to departments and employees are a challenging task. Optimal distribution of office space has the ability to maximizing synergies within an organization between staff. They study the efficiency of a greedy search algorithm and a tabu search algorithm in this article to generate high-quality alternatives to the issue of allocating office room.

3. METHODOLOGY

Tabu search and Ant colony optimization algorithms were both adapted to solve the problem of office space allocation and also test which of the two meta-heuristic algorithms has optimal solution in related to the criteria consider which are execution time and computer memory.

The tabu search algorithm is as followed:

STEP 1: Choose an initial solution i in S (solution space)
Set i* = initial solution I and parameters of iteration k = 0
STEP 2: Set k = k + 1 and generate a subset of V* of solution in N(i, k) such that either one of the tabu conditions is violated or at least one of the aspiration conditions holds
STEP 3: Choose a best j (i.e the best solution in the neighborhood solution) V* and set i = j
STEP 4: If F(i) < F(i*) then set i* = i
STEP 5: Update tabu list
STEP 6: If a stopping condition is met then stop.
Else go to step 2

Adaptation of Tabu-search algorithm to office space allocation.

1. Get list of lecturers and rooms to be allocated.
2. Initialize parameters i.e numbers of room, the maximum tabu size and the counter.
3. Loop through all available room
4. For each room, check for the appropriate lecturers that fit in for the current room.
5. The list of the lectures that fit in to the current room will be termed as sNeighborhood
6. Loop through the sNeighborhood, then for each lecturers in sNeighborhood check if they are not in the tabu list and check if they can be placed into the current room. If yes, then assign the lecturer to the current room.
7. Update tabu list
8. Check for the length of the current tabu list if not greater than the initialize value
9. If greater than then pop out the first element in the tabu list
10. Repeat step 3
11. Return sbest

```
procedure ACOMetaheuristicStatic
        Set parameters, initialize pheromone trails
        SCHEDULE_ACTIVITIES
                ConstructAntSolutions
                DaemonActions
                UpdatePheromones
        END_SCHEDULE_ACTIVITIES
end
```

Adaptation of Ant colony optimization to office space allocation

1. Get list of all lecturers and room to be allocated.
2. Initialize parameter Pheromone trial which indicate the path to start with, the shortest path
3. While room counts is greater than zero.
4. Generate solution by using the initial path to get all lecturers that fit into that path.
5. Check if there is pheromone in the path and if is string I,e if lecturer as not being assign to that room and the lecturer fit into that room.
6. Perform daemon action on each lecturer i.e assign lectures to that path i.e room.
7. Update pheromone i.e add lecturer to the path list.
8. Repeat step 5.

Several paths will be created by ants from their nest to the food destination in order to identify the shortest means of allocating a suitable office space to staffs, the shortest path can now be used to reach the destination. Once the shortest path has been found, all the ants will use the same path to move from starting point to end point. All the searching process of the random paths and allocation of the shortest path from starting point to ending point will be process internally by the algorithm. So as a result, the founded shortest path will be display as an output due to time efficiency.

Implementation

The research was implemented using visual (C#) to develop a console application as the frontend while Microsoft SQL Server was use for the backend. Modular programming approach was used in the implementation of this research work program. The modules are categorizing into two namely: Object definition (**Struct**) to determine all entities that partake in the program and Class definition was used for generic functions to perform special task in accomplishing the algorithm specifications.

4. RESULTS AND DISCUSSIONS

Table 1 for Tabu Search Algorithm run-time result

Test	Algorithm	Execution Time (sec)	Memory use (byte)
1	TS	0.8600458	23601152
2	TS	0.3113491	23650304
3	TS	0.6090045	23572480
4	TS	0.4083531	23568384
5	TS	0.6368304	23502848
6	TS	0.5231888	23617536
7	TS	0.4142562	23633920
8	TS	0.6048583	23646208
9	TS	0.7950324	23642112
10	TS	1.1213343	23605248

Table 1 shows the result of TABU algorithm in all the ten runs, considering the run time and memory used. The second run gives the best result, in term of run time of 0.3113491 seconds, 23650304 kilobytes. The tenth run returned the worse run time of 1.1213343 seconds and 23605248 kilobytes.

Table 2 for Ant colony optimization Algorithm run-time result

Test	Algorithm	Execution Time (sec)	Memory use (byte)
1	**ACO**	**2.0166519**	**23699456**
2	ACO	3.4968457	23736320
3	ACO	3.6566096	23658496
4	ACO	0.7818604	23625728
5	ACO	0.9608116	23597056
6	ACO	0.9467933	23728128
7	ACO	1.2283418	23670784
8	ACO	0.9253426	23699456
9	ACO	0.909971	23658496
10	ACO	1.3968302	23650304

Table 1 shows the result of Ant Colony optimization algorithm in all the ten runs considering the run time and memory used. The fourth run gives the best result, in term of run time of 0.7818604seconds, 23625728kilobytes. The third execution returned the worse run time of 3.6566096seconds and23658496kilobyte.

Tabu Search Performance On The Average:
Average performance = 0.8600458 + 0.3113491 + 0.6090045 + 0.4083531 + 0.6368304 + 0.5231888 + 0.4142562 + 0.6048583 + 0.7950324 + 1.1213343 =
6.2842529 / 10 = 0.62842529 on the average

Ant Colony Optimization Performance On The Average:
Average performance = 2.0166519 + 3.4968457 + 3.6566096 + 0.7818604 + 0.9608116 + 0.9467933 + 1.2283418 + 0.9253426 + 0.909971 + 1.3968302 =
16.3200581 / 10 = 1.63200581 on the average
As it can be seen from the above average evaluation, tabu search still smart perform ant colony optimization algorithm on the average performance.

5. SUMMARY AND CONCLUSION

The distribution of office space in any organization is generally a problematical issue, which regularly requests a significant measure of time to perform physically. This distribution's result affects the life of anyone who uses the office. Space has become an undeniably valuable commodity in higher education. In academic institutions, the distribution of the available room space to staff is a process that needs to be carried out on a regular basis because of the continuous changes that occur in this environment. Tabu search and ant colony optimization has been adapted to solve the problem of space allocation in academic institution and their performance has been evaluated in solving office space allocation problem. After series of iterations, it was discovered that tabu search smart performs the ant colony optimization algorithm in terms of average execution time of 0.62842529 as against Ants' colony 1.63200581(10 iterations). The benchmark instances proposed by the University Teaching Commission for tertiary institution were employed in the evaluation of the office space allocation problem. The research was implemented using visual C# to develop a console application as the frontend while Microsoft SQL Server was use for the backend.

However, according to no free lunch theorem of Wolpert and Macready (1995) it states that the averaged performance across all possible problems is the same for all algorithms. In other words, considering all possible problems, all algorithms perform equally and therefore, no distinction can be made between two algorithms because there are as many problems for which one algorithm performs better than the second one as for which the reverse is true.

6. RECOMMENDATION

The scope of this research work is to check for the performance evaluation of tabu search and ant colony optimization algorithm. This project uses a conventional language to implement the algorithm. However, other technologies like MATLAB, neural network and many more can be used to solve the space allocation problem. I therefore recommended for any researcher who might want to embark on this project to consider the aforementioned procedures and widen the scope of the space allocation problem.

REFERENCES

1. Adewumi, A .O, Ali, M. M, & Joshua. (2005). A Multi-level Genetic Algorithm for a Multi-stage Space Allocation Problem. Department of Computer Sciences, Faculty of Science, University of Lagos, Lagos, Nigeria.
2. Agboola, O.M (2018). Development of Hybrid Metaheuristic Algorithm for Solving NP-Hard Combinatoria Office Space Allocation problem in a Nigeria University, PhD Thesis, Computer Dept. University of Ilorin, Nigeria
3. Asaju, L. B, Ikechi, M & Peter, B. S. (2017). Optimization of office space allocation using artificial bee colony. Department of Computer Science, faculty of pure and applied sciences.
4. Asaju, L. B, Ikechi, M & Peter, B. S. (2019). Adaptation of late acceptance of hill climbing LAHC algorithm to tackle the (OFA) problem. Department of Computer Science, Faculty of Pure and Applied Sciences, Federal University Wukari, Wukari, Taraba State, Nigeria.
5. Burke and Varley (2013). Optimization of office space allocation. International Journal of Engineering Technology Science and Research
6. Burkard, R. E, &Rendl, F. (1984). A thermodynamically motivated simulation Procedure for combinatorial optimization problems. European Journal of Operations Research, 17(2), 169- 174.
7. Carvalho, J. M., (1999) Exact solution of bin packing problems using column generation and Branch and bound. Annals of Operations Research, 86, 629-659.
8. Chen, M, C & Lin, C.P. (2007). A data mining approach to product assortment and shelf space allocation. Expert Systems with Applications, 32, 976–986.
9. Landa, D. S, & Burke, E. K. (2004). Automating office space allocation issues in higher education. International Journal Engineering Technology Science and Research.
10. Garey, M. R, & Johnson, D. S. (1979). Computers and intractability: A guide to the theory of NP-Completeness.
11. Kellerer, H, &Pferschy, U. (1999). A new fully polynomial approximation scheme for the knapsack problem. Journal of combinatorial optimization 3:123-134. July, 2006
12. Lopes & Daniela (2015). The issue of office-space-allocation (OSA) for highly Hierarchical organizations. International Journal of Engineering Technology Science and Research
13. Martello, M, & Toth, P. (2001). Knapsack problems: algorithm and computer Implementations. University of Bologna, Bologna, Italy.
14. Mohammed, A. A, Ahamad, T. K, &Phuah, C, W. (2005). Harmony search for the office space allocation
15. problem.International Journal of Engineering Technology Science and Research.
16. Papadimitriou, C. H, &Steiglitz, K. (1999) Combinatorial optimization: Algorithms and Complexity.The International Journals of Engineering and sciences (IJENS).
17. Pereira, R, Kevin, C, & Rex, K. H. (2015). Office space allocation optimization.The international journal of computer science.
18. Poole, D, keith, A. M, & Randy, G. (1998). Computational intelligent: A logical Approach.The International Journals of Engineering and sciences (IJENS).
19. Ulker, O, & Dario, L. S. (2015). 0/1 integer programming model to address the problem of office space allocation. International Journal of Engineering Technology Science and Research
20. Wolpert, D, & Macready, W. (1995). No free lunch theorems for optimization problem IEEE transactions of evolutionary computation, vol.1, No 1, April, 1997.
21. Wren, A. (1990). Scheduling, timetabling and rostering – a special relationship. International conference on the practice and theory of Automated theory.
22. Yamada, T, &Futakawa, M. (1997). The knapsack sharing problem: An exact Algorithm. Journal of combinatorial optimization, march 2002, volume 6, issue 1, pp 35-54.

23. Yang, M. H. (2001). Retail shelf space allocation analysis using system dynamics approach.Journal of quality and technology management volume IX issue1 june 2013, page 99-129
24. Yang, Z. A, & Chen, W. C. (1999). A study on shelf space allocation and management. International journal of production economics, Elsevier, vol. 60(1), pages 309-317, April.
25. Zufryden, F. S. (1986). Joint optimization of retail price and shelf space.journal of retailing manuscript # MS 2008-MRP-014.
26. Ant colony optimization. (1992). About Ant system.Retrieved from http://wikipedia.com
27. No free lunch theorem. (1985) Retrieved fromhttp://www.scholarpedia.org
28. Optimization of office space allocation problem. (2017) Retrieved from https://scholar.google.co.uK
29. Tabu search. (1986). How tabu search algorithm work. Retrieved from http://www.google.com