NEW FORMULATION OF OPTIMAL FLOOR LAYOUT PLANNING OPTIMIZATION BY LOCALSOLVER

(株) セック　＊シンカロウ　ミカライ　SHYNKAROU Mikalai
01508070　東京大学　本間　裕大　HONMA Yudai

1. INTRODUCTION

The following is a proposal for optimizing the mathematical model proposed in "OPTIMIZATION MODEL FOR FLOOR LAYOUT PLANNING CONSIDERING ZONING" by Yuki Munemasa, Yudai Honma, and Ayumi Mukai. The solver used for creating and solving the model is LocalSolver.

2. MODEL OPTIMIZATION

The original model proposed by the study above showed a rapid increase in the time required to solve the optimization problem as the number of grids increased when using LocalSolver. Using such model to solve real life problems was very impractical. In order to improve the performance of the model, we have rewritten the model using the same idea as in the study above but using expressions that are optimized for LocalSolver. One of such improvements was using the built-in variable "list" that is highly optimized for splitting a set into several subsets and for sorting of sets. We demonstrate the usability of the "list" variable by showing already optimized models for solving "Traveling Salesman" problem and "Vehicle Routing" problem that use the "list" variable. And since the goal of the original study is to split a set of grids among a certain number of programs, the model written using the "list" variable would have much better performance compared to the original model. But since the model variables needed to be changed from Boolean variables to "list" variables, some of the original constraint conditions needed to be rewritten as well. And while most of the conditions, like each grid belonging to exactly one program and program area constraint, could be easily rewritten by using the "list" build-in functions like "partition" and "count", other were more difficult to adapt for the new model. In particular, spatial connectivity of grids constraint conditions needed to be completely rewritten. Instead of using links variables in the original model we have used the order property of the "list" variable to make it so that each grid in the "list" is adjacent to at least 1 other grid which appears earlier in the same "list". Doing so is like starting with a set containing only one grid and then progressively adding more grids to it one by one with the condition that each added grid needs to be adjacent to some grids that are already inside the set. Another notable change is the object function. In the original model the objective function was used to minimize the total length of "walls" between rooms (programs), while in the proposed model the objective function is used to maximize the total number of edges inside each room (program). And while these objective functions might appear very different, they both serve the same purpose of trying to make the shape of programs as close to square as possible.

3. MODEL COMPARISON

"Table 1" below shows floorplans that were used to compare the performance of both models and the amount of time that took both models to solve the optimization problem. All of the floorplans have identical shape, with the only difference being the number of grids into which the floorplan is being split.

	floorplan1	floorplan2	floorplan3	floorplan4	floorplan5	floorplan6	floorplan7
No. of grids	24	37	45	54	96	121	145
No. of edges	36	58	74	90	168	216	261
Time until solution (sec)	88→1	824→1	26505→2	864000> →28	?→976	?→1823	?→14334
Time until best solution (sec)	463→1	1535→4	39279→4	?→32	?→4518	?→6806	?→47374

Table 1

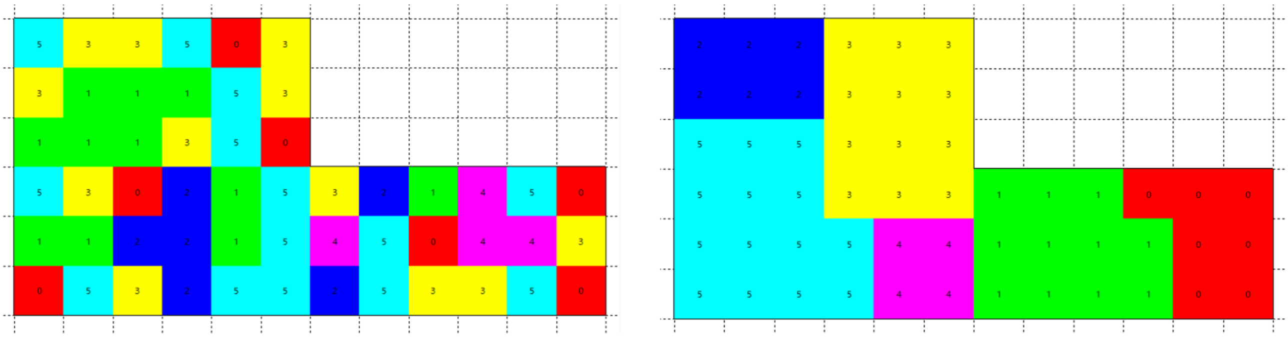
※ The time on the left of the arrow "→" shows the performance time of the original model, while the time on the right shows the performance time of the new model.

※ "Time until best solution" represents the timepoint at which the best solution for the model was acquired after evaluating the model for 1 day. (For floorplans "floorplan1" – "floorplan5" the best solution is also the optimal solution)

By using the original model, we were able to acquire a solution for floorplans "floorplan1" – "floorplan3", but for "floorplan4" and further we couldn't get a solution even after evaluating the model for 10 days.

The image below shows the best solutions for both models for "floorplan4" (original model on the left, new model on the right). You can see that the original model fails to get even an approximate solution after 10 days, while the new model calculates the optimal solution in a bit over half a minute.

The new model's performance starts to fall off after "floorplan6", and while it fails to find the optimal solution, it still manages to find a solution that satisfies all the restrictions of the problem.



floorplan4

4. CONCLUSION

The above comparison shows that the new model is much more efficient at finding solutions when using LocalSolver. And while this model also starts to fall off as the number of grids increases, it starts falling off much later and at a lower speed compared to the original model.

References

[1]Yuki Munemasa, Yudai Honma, and Ayumi Mukai: OPTIMIZATION MODEL FOR FLOOR LAYOUT PLANNING CONSIDERING ZONING, International Symposium on Scheduling 2019, July 5-7, Matsue, Japan, 2A1

[2]https://www.localsolver.com/: Optimization Model Solver

[3]https://www.localsolver.com/docs/last/exampletour/tsp.html: "Traveling Salesman" problem model

[4]https://www.localsolver.com/docs/last/exampletour/vrp.html: "Capacitated Vehicle Routing" problem model