THE INSTITUTE FOR SYSTEMS RESEARCH

ISR TECHNICAL REPORT 2014-04

Modeling Separations of Plant Layout Problems

Jeffrey W. Herrmann

Modeling Separations of Plant Layout Problems

Jeffrey W. Herrmann
A. James Clark School of Engineering
University of Maryland
College Park, MD 20742
301-405-5433
jwh2@umd.edu

Abstract

This paper describes the results of a simulation study that evaluated the performance of different separations of the plant layout problem solved by bounded rational decision-makers. Seven problem instances from the literature were studied. We simulated the solution of a problem by a bounded rational decision-maker as a random search over the solution space. The problem was separated by identifying "subsets" of adjacent locations. The subset assignment problem partitioned the departments into subsets corresponding to these subsets of locations. Then, the subset layout problem assigned the locations in the subset to the departments. We considered separations with 2, 3, and 4 subsets. We also considered separations that first aggregated the departments before assigning them to subsets of locations. The results showed that separating the problem can lead to better solutions than solving the problem all-at-once, but some separations lead to worse solutions. Maximizing the flow inside the subsets generated better solutions than maximizing the adjacency of the departments inside the subsets. When fewer subsets are used, minimizing the cost inside each subset generated better solutions than minimizing the total cost. These results show that the quality of the solutions created by a design process is influenced by the choice of subproblems that make up the design process.

Introduction

The plant layout problem is a difficult combinatorial optimization problem that can occur during the design (or redesign) of a factory, a hospital, or other facility through which entities (e.g. parts and patients) flow. The primary objective is to minimize the cost of moving these entities by assigning different functions (departments) to the locations available.

The plant layout problem (also known as the facility layout problem) is NP-complete and can be formulated as a quadratic assignment problem (Koopmans and Beckman, 1957). Kusiak and Heragu (1987) reviewed formulations of the problem and compared the performance of different construction heuristics and improvement algorithms.

In practice, human decision-makers often separate a complex optimization problem like the plant layout problem into subproblems and then solve each subproblem instead of tackling the complete problem. This approach is a natural strategy given the constraints that human decision-makers have.

The performance of humans on the plant layout problem has been previously studied. Although Scriabin and Vergin (1975) described an experiment in which human subjects generated solutions that were better than those constructed by layout heuristics, Trybus and Hopkins (1980) conducted an experiment in which the solutions generated by human subjects were not better than those generated by the CRAFT algorithm.

If human decision-makers were able to optimize, then separating a problem into subproblems would usually lead to solutions that are inferior to those found by solving the problem all-at-once (only in certain conditions will the optimal solutions to the subproblems form an optimal solution to the complete problem).

It is well-known, however, that real-world decision-makers cannot optimize because of limits on their problem-solving capacity. This concept is known as “bounded rationality” (Simon, 1997a). Bounded rationality reflects the observation that, in most real-world cases, decision-makers have limited information and limited computational capabilities for finding and evaluating alternatives and choosing among them (Simon, 1997b; Gigerenzer *et al.*, 1999; March and Simon, 1993). A decision-maker cannot perfectly evaluate the consequences of the available choices. This prevents complete and perfect optimization.

The study described in this paper was motivated by the following questions: (1) How can the plant layout problem be separated? (2) Which separations of the plant layout problem generate better solutions than solving the problem all-at-once?

As part of an ongoing study of the effectiveness of separation by human decision-makers, this paper presents the results of a study that considered specific instances of the plant layout problem and simulated different separations of the problem using models of the searches of bounded rational decision-makers.

Note that these separations are not meant to improve upon state-of-the-art approaches for solving the plant layout problem; they are meant as models of the design processes used by bounded rational decision-makers.

Hong and Page (2004) modeled problem-solvers of limited ability as searches that they called "heuristics," and each heuristic searched a finite set of solutions until it could not find a better solution. Thus, the problem-solver is conducting a type of hill-climbing search. Hong and Page studied teams of such problem-solvers and identified conditions under which a diverse set of problem-solvers will likely perform better than a team of high-performing individuals. In their work, the problem-solvers searched a finite set of solutions (the size ranged from 200 to 10,000). A problem-solver was characterized by how many and which points near the current solution it would consider. Essentially, different problem-solvers had different neighborhood definitions.

LiCalzi and Surucu (2012) studied teams of problem-solvers in which different problem-solvers had different partitions of the solution space, which affected the team's ability to find the optimal solution. They considered how the size of the search space affected the size of the team needed to find the optimal solution.

Herrmann (2010) presented a method for assessing the quality of a product design process by measuring the profitability of the product that the process generates. Because design decision making is a type of search, the method simulated the choices of a bounded rational designer for each subproblem using search algorithms. The searches, which were limited to a fixed number of iterations, had random components (either randomly selecting a solution or randomly moving to a point near the existing solution) and a procedural structure to keep track of the best solution found. The results showed that decomposing a problem into subproblems yields a better solution than solving the entire problem at once when bounded rational search is employed. This result suggests that well-designed progressive design processes are the best way to generate profitable product designs.

Herrmann (2012) described a study in which different approaches for separating the Inventory Slack Routing Problem (a complex vehicle routing problem) were simulated. Again, a random search was used to simulate how a bounded rational human decision-maker would solve each subproblem. The results showed that the structure of the separation and the objectives used in each subproblem significantly affected the quality of the solutions that are generated. Additional details about the Inventory Slack Routing Problem can be found in the report by Montjoy and Herrmann (2012).

The remainder of the paper proceeds by formulating the plant layout problem, introducing the separations, and the describing the simulation models. The paper then presents the results of the study before concluding with some insights gained from this work.

The Plant Layout Problem

Let n be the number of departments in the plant. All of the departments are the same size, and there are n locations for the departments. A feasible solution (a layout) specifies a location for each department such that every department is assigned exactly one location and every location is assigned to exactly one department. Let c_{jl} be the cost of transporting one unit of material (one entity) from location j to location l (this can also be the distance from location j to location l). Let f_{ik} be the number of units of material (number of entities) that need to be transported from department i to department k (over a given time horizon). Let x_{ij} = 1 if department i is placed at location j (location j is assigned to department i) and 0 otherwise.

The plant layout problem can be formulated as follows:

$$\min \sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n} f_{ik} c_{jl} x_{ij} x_{kl}$$

$$\sum_{j=1}^{n} x_{ij} = 1,\ i = 1,\ldots,n$$

$$\sum_{i=1}^{n} x_{ij} = 1,\ j = 1,\ldots,n$$

Separations

A separation is a process that solves a sequence of subproblems. A large problem is divided into subproblems, and the solution to one subproblem provides the inputs to one or more subsequent subproblems. Note that the separation does not have to be a simple sequence of subproblems; it may have subproblems that are solved in parallel at places. A given separation specifies a partial order in which the subproblems are solved. A different order of subproblems would be a different separation and would lead to a different solution.

The subproblems' objective functions are surrogates for the original problem's objective function. These surrogates come from substituting simpler performance measures that are correlated with the original one, eliminating components that are not relevant to that subproblem, or from removing variables that will be determined in another subproblem.

We considered separations with "subsets" of adjacent locations. Let N be the number of subsets. Let S_a be subset a, $a = 1, \ldots, N$. The subsets are mutually exclusive and collectively exhaustive. Thus, $\bigcup_{a=1}^{N} S_a = \{1,\ldots,n\}$. Each subset is an aggregate location in which multiple departments can be placed.

We also considered separations with "aggregate departments" that combine multiple departments. The aggregate departments are mutually exclusive and collectively exhaustive. Each aggregate department is placed in a subset of the appropriate size.

The first subproblem is the subset assignment problem. Each and every department must be assigned to exactly one subset. The costs between locations were ignored. Two versions of the problem were considered. The first version (SA1) considered only the existence of a flow (a "connection") between departments and maximizes the number of connections between departments in the same subset. Let A_{ik} = 1 if f_{ik} > 0 and 0 otherwise. Let y_{ia} =1 if department i is assigned to subset a and 0 otherwise.

$$\max \sum_{a=1}^{N} \sum_{i=1}^{n} \sum_{k=1}^{n} A_{ik} y_{ia} y_{ka}$$

$$\sum_{a=1}^{N} y_{ia} = 1,\ i = 1,\ldots,n$$

$$\sum_{i=1}^{n} y_{ia} = |S_a|,\ a = 1,\ldots,N$$

The second version of first subproblem (SA2) maximizes the flow inside the subsets.

$$\max \sum_{a=1}^{N} \sum_{i=1}^{n} \sum_{k=1}^{n} f_{ik} y_{ia} y_{ka}$$
$$\sum_{a=1}^{N} y_{ia} = 1, \; i = 1, \ldots, n$$
$$\sum_{i=1}^{n} y_{ia} = |S_a|, \; a = 1, \ldots, N$$

Note that neither SA1 nor SA2 considers c_{jl}, the unit transportation cost (distance). The grouping of the departments and the placing of these groups are done simultaneously. SA3 and SA4 take a different approach by first grouping (aggregating) the departments and then determining their locations. In order to do this, all of the subsets must be the same size. That is, $|S_a| = n / N$ for $a = 1, \ldots, N$.

SA3 begins by solving SA2 to group the departments into *N* "aggregate departments." It then determines the best subset for each aggregate department. Let G_b be the set of departments assigned to aggregate department *b*, *b* = 1, …, *N*. (That is, y_{ib} =1 for all $i \in G_b$.) Determine $\hat{f}_{bc}$, the aggregate flow between aggregate departments *b* and *c*, and $\bar{c}_{ad}$, the average cost between subsets *a* and *d*, as follows:

$$\hat{f}_{bc} = \sum_{i \in G_b} \sum_{k \in G_c} f_{ik}$$
$$\bar{c}_{ad} = \frac{N^2}{n^2} \sum_{j \in S_a} \sum_{l \in S_d} c_{jl}$$

Let $\hat{x}_{cd}$ = 1 if aggregate department *c* is placed at subset *d* (subset *d* is assigned to aggregate department *c*) and 0 otherwise. (If $\hat{x}_{cd}$ = 1, all of the departments in G_c are placed in subset *d*.) Then, SA3 minimizes the total aggregate cost, a problem that can be formulated as follows:

$$\min \sum_{b=1}^{N}\sum_{c=1}^{N}\sum_{a=1}^{N}\sum_{d=1}^{N} \hat{f}_{bc}\bar{c}_{ad}\hat{x}_{ba}\hat{x}_{cd}$$

$$\sum_{d=1}^{N} \hat{x}_{cd} = 1,\ c = 1,\ldots,N$$

$$\sum_{c=1}^{N} \hat{x}_{cd} = 1,\ d = 1,\ldots,N$$

SA4 begins by using a greedy heuristic (described in Appendix B) to group the departments into N "aggregate departments." It then determines the best subset for each aggregate department in the same way that SA3 does.

Let D_a be set of departments placed in subset a, a = 1, …, N. (From SA1 and SA2, department i is an element of D_a if and only if y_{ia} =1. From SA3 and SA4, department i is an element of D_a if and only if i is an element of G_b and $\hat{x}_{ba}$ = 1.)

Given subset assignments (a solution to SA1, SA2, SA3, or SA4), the departments must be assigned locations from their subsets. This subproblem can be solved with all of the subsets (LA1) or with each subset individually (LA2).

The subproblem LA1, which minimizes the total cost (including the cost of flow between subsets), can be formulated as follows:

$$\min \sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n} f_{ik}c_{jl}x_{ij}x_{kl}$$

$$\sum_{j\in S_a} x_{ij} = 1,\ a = 1,\ldots,N; i \in D_a$$

$$\sum_{i\in D_a} x_{ij} = 1,\ a = 1,\ldots,N; j \in S_a$$

The subproblem LA2(a), which minimizes the cost within subset a, can be formulated as follows:

$$\min \sum_{i\in D_a}\sum_{k\in D_a}\sum_{j\in S_a}\sum_{l\in S_a} f_{ik}c_{jl}x_{ij}x_{kl}$$

$$\sum_{j\in S_a} x_{ij} = 1,\ i \in D_a$$

$$\sum_{i\in D_a} x_{ij} = 1,\ j \in S_a$$

Thus, for the given subsets, four separations are possible: SA1-LA1, SA2-LA1, SA1-LA2, SA2-LA2. The separations with LA1 have two subproblems; the separations with LA2 have 1 + *N* subproblems. Note that the *N* instances of LA2 could be solved in parallel.

If the subsets are the same size, then four other separations are possible: SA3-LA1, SA3-LA1, SA4-LA2, and SA4-LA2.

Modeling Searches

An important aspect of bounded rationality is that the resources and time available for problem-solving are limited. Consequently, the proposed model of a bounded rational decision-maker incorporates limits that will constrain the amount of time available for the search.

We used a search algorithm that identifies and evaluates solutions to model the choices of a bounded rational decision-maker. At each iteration, the search randomly selects a solution. The procedural structure of the model attempts to compensate for this randomness by keeping track of the best solution found so far. Finally, the search is limited to a fixed number of solutions.

Because the plant layout problem is a type of combinatorial optimization problem, the search randomly generates a permutation of the departments in order to generate a solution. (For LA1 and LA2, the search generates a random permutation of the departments in one subset.) Let T be the search effort. Let $f(X)$ be the objective function.

The random sampling search works as follows: Do the following step T times: Randomly select a solution X. If $f(X)$ is the best function evaluation found so far, keep X as the best solution found so far.

This search returns X, the "best" solution found so far.

It is important to note that this search is meant to represent a bounded rational decision-maker. It should not be compared to state-of-the-art techniques for solving plant layout problems. Like Gurnani and Lewis (2008) and Herrmann (2010, 2012), we are modeling the decision-maker's bounded rational design choices as a random process. To model a bounded rational designer who is using a type of fast and frugal heuristic, these searches have a simple rule to stop the search (when the number of solutions evaluated equals *T*) and to choose a solution (whether it is better than the best found so far).

Instances

To compare the performance of different separations, we used seven plant layout problems from the literature. The instances SCR12, SCR15, SCR20, ELS19, KRA30a, KRA30b, and KRA32 were downloaded from Burkard *et al.* (2012/2014), which also listed optimal solutions for these problems. These instances of the QAP are based on facility layout problems. In these instances, the flow from department *i* to department *k* equals the flow from department *k* to department *i*.

The costs (distances) between locations in the SCR12, SCR15, and SCR20 correspond to rectangular grid layouts (Scriabin and Vergin, 1975), which made identifying subsets easy. We simply partitioned the locations based on their location in these layouts (shown in Figure 1.)

For the 12-location instance, we created a partition with four subsets: {1, 2, 5}, {6, 9, 10}, {3, 4, 7}, and {8, 11, 12}. We combined the first two subsets and the last two subsets to create a partition with two subsets. These partitions we labeled "3333" and "66." (Each number in a label refer to the number of locations in each subset.)

For the 15-location instance, we created a partition with four subsets: {1, 2, 5, 6}, {3, 4, 7, 8}, {9, 10, 13, 14}, and {11, 12, 15}. We combined the first two subsets and the last two subsets to create a partition with two subsets. These partitions we labeled "4443" and "87."

For the 20-location instance, we created a partition with four subsets: {1, 2, 5, 6, 9, 10}, {3, 4, 7, 8, 11, 12}, {13, 14, 17, 18}, and {15, 16, 19, 20}. We combined the last two subsets to create a partition with three subsets. We then combined the first two subsets to create a partition with two subsets. These partitions we labeled "6644," "668," and "128."

According to Elshafei (1977), the ELS19 instance was based on a real hospital, but little information about the spatial layout is given (beyond the fact that locations were on different floors). We created subsets as follows: let the graph $G = (N, E)$ be a graph where N is a set of nodes that correspond to the locations, and E is the set of all edges between different nodes. Then, we then eliminated all edges (j, l) with $c_{jl} > 55$. This yielded three connected subgraphs, and we designated the locations corresponding to the nodes in each subgraph as a subset (shown in Table 1). This partition we labeled "298."

Then, we then eliminated all edges (j, l) with $c_{jl} > 40$. This yielded 4 connected subgraphs, and we designated the locations corresponding to the nodes in each subgraph as a subset (shown in Table 1). This partition we labeled "2953."

For the KRA30a, KRA30b, and KRA32 instances (Krarup and Pruzan, 1978), we exploited the structure of the instance to create subsets of locations as follows: first, we identified four subsets in which the distance between every pair of locations in the same subset was less than or equal to 200 units; then, we identified two subsets in which the distance between every pair of locations in the same subset was less than or equal to 300 units (which were unions of the smaller subsets). For the KRA30b instance, we identified three subsets in which the distance between every pair of locations in the same subset was less than or equal to 265 units (this combined the two subsets with only six locations). Table 2 lists the partitions and their subsets.

For testing the separations that include SA3 and SA4, we considered the SCR12, SCR20, and KRA32 instances because their symmetry facilitated creating subsets of equal sizes. The locations in the SCR12 instance were divided into six subsets (each with two locations), four subsets (each with three locations), and three subsets (each with four locations). The locations in the six subsets were {1, 2}, {3, 4}, …, {11, 12}. The locations in the four subsets were {1, 2, 5}, {6, 9, 10}, {3, 4, 7}, and {8, 11, 12}. The locations in the two subsets were {1, 2, 5, 6, 9, 10} and {3, 4, 7, 8, 11, 12}. These are the same as the 3333 and 66 partitions previously mentioned for this instance.

The locations in the SCR20 instance were divided into ten subsets (each with two locations), five subsets (each with four locations), and two subsets (each with ten locations). The locations in the ten subsets were {1, 2}, {3, 4}, …, {19, 20}. The locations in the five subsets were {1, 2, 5, 6}, {3, 4, 7, 8}, {9, 10, 13, 14}, {11, 12, 15, 16} and {17, 18, 19, 20}. The locations in the two subsets were {1, 2, 5, 6, 9, 10, 13, 14, 17, 18} and {3, 4, 7, 8, 11, 12, 15, 16, 19, 20}.

The locations in the KRA32 instance were divided into 16 subsets (each with two locations), eight subsets (each with four locations), four subsets (each with eight locations), and two subsets (each with 16 locations). The locations in the 16 subsets were {1, 2}, {3, 4}, …, {31, 32}. The locations in the eight subsets were {1, 2, 3, 4}, …, {29, 30, 31, 32}. The locations in the four subsets were {1, …, 8}, {9, …, 16}, {17, …, 24}, and {25, …, 32}. The locations in the two subsets were {1, …, 16} and {17, …, 32}. These are the same as the 8888 and 1616 partitions previously mentioned for this instance.

Table 1. The subsets of locations for the ELS19 instance.

Partition	Subsets
2953	{15, 16} {7, 8, 9, 11, 12, 14, 17, 18, 19} {1, 2, 3, 4, 6}, {5, 10, 13}
298	{15, 16}. {7, 8, 9, 11, 12, 14, 17, 18, 19}, {1, 2, 3, 4, 5, 6, 10, 13}

Table 2. The subsets of locations for the KRA30a, KRA30b, and KRA32 instances.

Instance	Partition	
KRA30a	7788	{1, …, 7}, {8, …, 14}, {15, …, 22}, {23, …, 30}
	1416	{1, …, 14}, {15, …, 30}
KRA30b	9696	{1, …, 9}, {10, …, 15}, {16, …, 24}, {25, …, 30}
	9912	{1, …, 9}, {16, …, 24}, {10, …, 15, 25, …, 30}
	1515	{1, …, 15}, {16, …, 30}
KRA32	8888	{1, …, 8}, {9, …, 16}, {17, …, 24}, {25, …, 32}
	1616	{1, …, 16}, {17, …, 32}

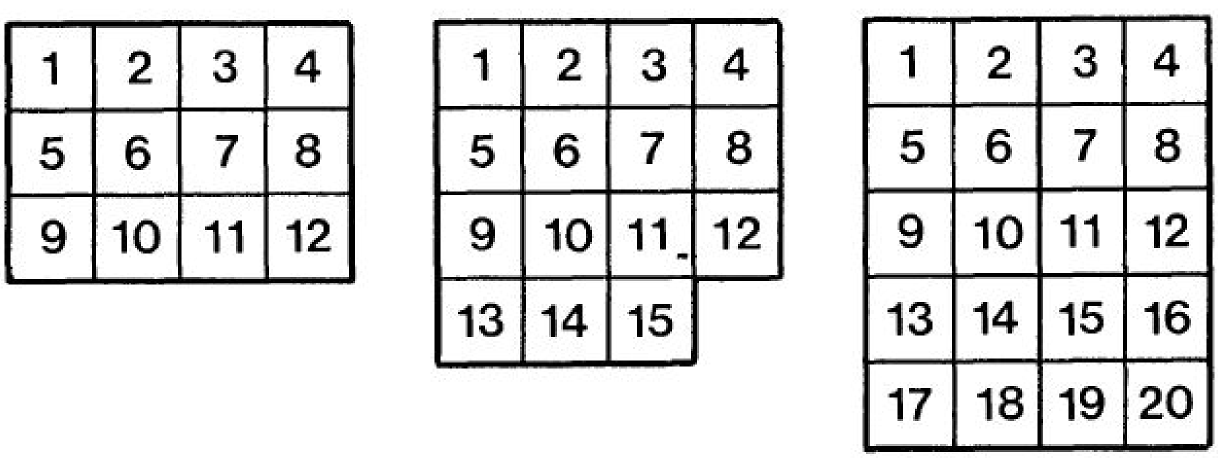

Figure 1. Plant layouts for the 12-, 15-, and 20-location instances from Scriabin and Vergin (1975).

Computational Experiments

The purpose of the computational experiments was to compare the performance of the separations. We considered the seven instances described in the previous section.

For each instance, we tested the all-at-once search and the four separations (SA1-LA1, SA2-LA1, SA1-LA2, SA2-LA2) using the relevant partitions. Thus, for the SCR12, SCR15, ELS19, KRA30a, and KRA32 instances, which had two partitions, there were 8 separations. For the SCR20 and KRA30b instances, which had three partitions, there were 12 separations.

We then tested the SA2-LA1, SA2-LA2, SA3-LA1, SA3-LA2, SA4-LA1, and SA4-LA2 separations on the SCR12, SCR20, and KRA32 instances using the subsets mentioned previously. (The SA1- separations were not used because the first set of results showed that they generated worse solutions.)

A solution to a subproblem was found by a search that randomly generated 2000 feasible solutions and kept the best one generated. The all-at-once search and the separations were all run 1000 times. The total cost was used to measure the quality of a solution.

Results

Here we report the average total cost of the solutions generated by the all-at-once search and the separations. We calculated confidence intervals on the mean difference and determined that the differences were statistically significant. Tables 3 to 7 provide the sample means over the 1000 replications, and Appendix A includes a table with the sample standard deviations of the results.

The results show that relative performance of the all-at-once and separation approaches varied across the partitions considered. The relative performance of the approaches on the ELS 19 instance was not affected by the partition (because they are very similar).

The SA2- separations performed better than the SA1- separations for all of the instances and partitions. Maximizing the flow within the subsets led to better subset assignments than simply maximizing the connections within the subsets. On the KRA30a, KRA30b, and KRA32 instances, however, the performance of the SA1- separations was closer to the performance of the SA2- separations because the positive values of f_{ik} in these instances had the same magnitude (the only positive values were 1, 2, 3, and 4); thus, maximizing connections was approximately equivalent to maximizing flow. Hereafter, we will focus on the SA2- separations.

The SA2-LA1 separations performed better than the all-at-once search in the cases considered except for the 3333 partition in the SCR12 instance. The LA1 subproblem has a smaller space of solutions (because the departments have already been assigned to subsets of locations), and it minimizes the total cost, which yields good solutions.

The SA2-LA2 separations performed worse than the all-at-once search for the partitions with four subsets (the 3333, 4443, and 6644 partitions) in the SCR12, SCR15, and the SCR20 instances. In the subset assignments for these partitions, only 51% to 58% of the total flow was inside the subsets (i.e., between departments in the same subset), so ignoring the flow between subsets (as the LA2 subproblem does) led to poor solutions. For the partitions with only two subsets (the 66, 87, and 128 partitions), however, 80% to 88% of the total flow was inside the subsets, so ignoring the flow between subsets was not a handicap, and the SA2-LA2 separations performed better than the all-at-once search and the SA2-LA1 separations.

On the ELS19 instance, the SA2-LA2 separation performed better than the all-at-once search and the SA2-LA1 separation for both partitions. For the partitions for this instance, 86% to 88% of the total flow was inside the subsets found when solving the SA2 subproblem, so ignoring the flow between subsets was not a handicap.

On the KRA30a, KRA30b, and KRA32 instances, the total flow inside the subsets was lower: only 33% to 35% with four subsets (the 7788, 9696, and 8888 partitions), 44% with three subsets (the 9912 partition), and only 61% to 63% with only two subsets (the 1416, 1515, and 1616 partitions). The quality of the solutions generated by the SA2 separations was better than the solutions generated by the all-at-once search for the partitions with only two subsets. For the partitions with three or four subsets, the SA2-LA1 separation (which considers the total cost) generated better solutions than the all-at-once search and the SA2-LA2 separation. (The quality of the solutions generated by the SA2-LA2 separation was not significantly different from the quality of the solutions generated by the all-at-once search in the 7788 and 9696 partitions.)

Table 8 summarizes the comparison of the SA2-LA2 separation to the all-at-once search by instance and partition. The relative change in solution quality is the difference in the average total cost of the solutions generated by the SA2-LA2 separation and the average total cost of the solutions generated by the all-at-once search, divided by the optimal total cost for that instance. These show that, in the partitions with fewer subsets of locations, which had more flow inside the subsets, the SA2-LA2 separation generated better solutions.

The SA3- and SA4- separations, which formed the aggregate departments and then placed them in subsets, generated better solutions that the SA2- separations when there are more than two subsets. For the SCR 12 instance, the difference is not statistically significant when there are only two subsets. For the SCR 20 instance, when there are only two subsets, the SA3- and SA2- separations are not significantly different, but the SA4- separations are slightly worse. For the KRA 32 instance, when there are only two subsets, the SA3- and SA2- separations are not significantly different, but the SA4- separations are significantly better.

On the SCR 20 and KRA 32 instances, the SA4- separations, which used a greedy heuristic to form the aggregate departments, generated better solutions that the SA3- separations when there are more than two subsets. The aggregate departments constructed by greedy heuristic had more flow inside the subsets than those found by solving SA2. On the SCR 20 instance, the SA4-LA1 separation with 10 subsets generated the best solutions (on average); on the KRA 32 instance, the SA4-LA1 separation with 8 subsets generated the best solutions (on average).

Table 3. Average solution quality for the SCR 12 instance. Underlined numbers represent approaches that generated solutions better than the all-at-once approach.

Separation		Partition 3333	Partition 66
All-at-once	37,067		
SA1-LA1		39,236	36,068
SA2-LA1		37,680	33,462
SA1-LA2		50,945	43,131
SA2-LA2		43,564	33,273

Table 4. Average solution quality for the SCR 15 instance. Underlined numbers represent approaches that generated solutions better than the all-at-once approach.

Separation		Partition 4443	Partition 87
All-at-once	68,136		
SA1-LA1		70,155	62,367
SA2-LA1		64,636	61,177
SA1-LA2		87,871	63,908
SA2-LA2		74,730	57,977

Table 5. Average solution quality for the SCR 20 instance. Underlined numbers represent approaches that generated solutions better than the all-at-once approach.

Separation		Partition 6644	Partition 668	Partition 128
All-at-once	81,807			
SA1-LA1		80,876	77,343	75,756
SA2-LA1		75,128	72,879	72,583
SA1-LA2		94,274	88,787	79,555
SA2-LA2		84,090	77,579	71,555

Table 6. Average solution quality for the ELS 19 instance.
Underlined numbers represent approaches that
generated solutions better than the all-at-once approach.

Separation		Partition 2953	Partition 298
All-at-once	28,169,466		
SA1-LA1		33,575,662	33,997,103
SA2-LA1		21,837,025	22,089,271
SA1-LA2		38,300,240	40,192,981
SA2-LA2		21,145,597	20,482,920

Table 7. Average solution quality for the KRA30a, KRA30b, and KRA32 instances.
Underlined numbers represent approaches that
generated solutions better than the all-at-once approach.

Separation	Number of subsets	KRA30a	KRA30b	KRA32
All-at-once		118,666	120,525	120,487
SA1-LA1	4	115,909	118,066	117,949
SA2-LA1		115,217	116,820	116,863
SA1-LA2		119,376	122,219	120,844
SA2-LA2		118,520	120,574	119,505
SA1-LA1	3		116,907	
SA2-LA1			116,166	
SA1-LA2			122,960	
SA2-LA2			121,641	
SA1-LA1	2	113,674	115,265	115,030
SA2-LA1		112,535	114,069	113,854
SA1-LA2		113,603	115,215	114,732
SA2-LA2		112,003	113,647	112,984

Table 8. Average relative difference in solution quality for the SA2-LA2 separation and All-at-once search. (A negative number indicates that the SA2-LA2 separation generated worse solutions.)

Instance	Number of subsets	Partition	Relative flow inside the subsets (%)	Relative improvement in solution quality (%)
KRA32	4	8888	33	1.1
KRA30a	4	7788	33	0.2
KRA30b	4	9696	35	-0.1
KRA30b	3	9912	44	-1.2
KRA30b	2	1515	61	7.5
KRA32	2	1616	62	8.5
KRA30a	2	1416	63	7.5
SCR20	4	6644	51	-4.2
SCR15	4	4443	55	-12.9
SCR12	4	3333	58	-20.7
SCR20	3	668	61	7.7
SCR20	2	128	80	18.6
SCR15	2	87	84	19.9
SCR12	2	66	88	12.0
ELS19	4	2953	86	40.8
ELS19	3	298	88	44.7

Table 9. Average solution quality for the SCR 12 instance. Underlined numbers represent approaches that generated solutions better than the all-at-once approach.

	Number of subsets		
Separation	6	4	2
SA2-LA1	40,234	37,458	<u>33,411</u>
SA2-LA2	44,417	43,258	<u>33,203</u>
SA3-LA1	<u>35,495</u>	<u>34,485</u>	<u>33,456</u>
SA3-LA2	38,126	40,534	<u>33,197</u>
SA4-LA1	<u>35,156</u>	<u>35,673</u>	<u>33,461</u>
SA4-LA2	37,460	41,472	<u>33,217</u>

Table 10. Average solution quality for the SCR 20 instance. Underlined numbers represent approaches that generated solutions better than the all-at-once approach.

	Number of subsets		
Separation	10	5	2
SA2-LA1	88,940	<u>77,633</u>	<u>73,535</u>
SA2-LA2	97,880	88,141	<u>71,188</u>
SA3-LA1	<u>68,367</u>	<u>69,214</u>	<u>73,498</u>
SA3-LA2	<u>76,602</u>	<u>79,406</u>	<u>70,893</u>
SA4-LA1	<u>63,321</u>	<u>63,641</u>	<u>74,633</u>
SA4-LA2	<u>69,984</u>	<u>72,846</u>	<u>72,528</u>

Table 11. Average solution quality for the KRA 32 instance. Underlined numbers represent approaches that generated solutions better than the all-at-once approach.

	Number of subsets			
Separation	16	8	4	2
SA2-LA1	125,792	122,022	<u>116,975</u>	<u>113,916</u>
SA2-LA2	132,490	126,571	<u>119,773</u>	<u>113,363</u>
SA3-LA1	<u>111,342</u>	<u>110,017</u>	<u>113,642</u>	<u>113,935</u>
SA3-LA2	<u>118,071</u>	<u>114,617</u>	<u>116,427</u>	<u>113,466</u>
SA4-LA1	<u>106,172</u>	<u>104,071</u>	<u>105,309</u>	<u>108,781</u>
SA4-LA2	<u>110,215</u>	<u>108,243</u>	<u>105,764</u>	<u>108,647</u>

Summary and Conclusions

This paper presented the results of a computational study of different separations of the plant layout problem under the condition of bounded rational decision-makers. The problem was separated into two subproblems: (1) assign the departments to subsets of neighboring locations and (2) assign locations within these subsets to those departments. Random searches were used to represent the attempts of bounded rational decision-makers to solve these subproblems.

The results show that, on average, some separations generated better solutions than solving the problem all-at-once, and other separations did not. The best separations used only two subsets and maximized the flow inside the subsets for the first subproblem. Minimizing the total cost (which required considering all of the subsets at the same time) generated better solutions than solving the problem all-at-once. The quality of the solutions generated by minimizing the cost within the subsets (ignoring the flow between subsets) varied. When more subsets of locations were used and the total flow within the subsets was low (less than 60% of the total), this separation generated worse solutions than solving the problem all-at-once. When fewer subsets were used and the total flow within the subsets was high (more than 60% of the total), this separation generated better solutions than solving the problem all-at-once. Grouping departments based on connections generated worse solutions than grouping based on the magnitude of the flows between them.

These results indicate that, for these instances of the plant layout problem, separation leads to better solutions for bounded rational decision-makers. For some of these instances, using only two subsets generated the best solutions on average. For others, using more subsets and aggregating the departments first generated the best solutions.

These results reinforce the conclusions of Herrmann (2010) about the usefulness of separating complex optimization problems for bounded rational decision-makers and the importance of choosing the right objective function for the subproblems. It also demonstrated the usefulness of aggregation to generate smaller instances that can be used to find high-quality solutions quickly.

Additional research should consider larger instances and instances in which forming manufacturing cells is an option. Studies of how the allocation of decision-making resources affects solution quality would be valuable.

References

Burkard, R.E., E. Çela, S.E. Karisch, and F. Rendl, "QAPLIB - A Quadratic Assignment Problem Library," http://anjos.mgi.polymtl.ca/qaplib//inst.html, April 20, 2012, accessed July 19, 2014.

Elshafei, A.N., "Hospital layout as a quadratic assignment problem," *Operations Research Quarterly*, 28:167-179, 1977.

Gigerenzer, Gerd, Peter M. Todd, and the ABC Research Group, 1999, *Simple Heuristics That Make Us Smart*, Oxford University Press, New York.

Gurnani, Ashwin, and Kemper Lewis, 2008, "Collaborative, Decentralized Engineering Design at the Edge of Rationality," *Journal of Mechanical Design*, 130(12), 121101.

Herrmann, Jeffrey W., "Progressive Design Processes and Bounded Rational Designers," *Journal of Mechanical Design*, August 2010, Volume 132, Issue 8, 081005 (8 pages). doi:10.1115/1.4001902

Herrmann, Jeffrey W., "Separating the Inventory Slack Routing Problem," Technical Report 2012-8, Institute for Systems Research, University of Maryland, College Park, July, 2012. Available online at http://hdl.handle.net/1903/12865.

Hong, Lu, and Scott E. Page, "Groups of diverse problem solvers can outperform groups of high-ability problem solvers," *Proceedings of the National Academy of Sciences of the United States of America*, Volume 101, Number 46, pages 16385-16389, 2004.

Koopmans, T.C., and M. Beckman, "Assignment problems and the location of economic activities," *Econometrica*, Volume 25, pages 53-76, 1957.

Krarup, Jakob, and Peter Mark Pruzan, "Computer-aided layout design," in *Mathematical Programming in Use*, M.L. Balinski, editor, pages 75-94, North-Holland Publishing Company, Amsterdam, 1978.

Kusiak, Andrew, and Sunderesh S. Heragu, "The facility layout problem," *European Journal of Operational Research*, Volume 29, Number 3, pages 229-251, 1987.

March, James, and Herbert Simon, 1993, *Organizations*, second edition, Blackwell, Cambridge, Massachusetts.

LiCalzi, M., and O. Surucu, "The power of diversity over large solution spaces," *Management Science*, Vol. 58, No. 7, pp. 1408-1421, 2012.

Montjoy, Adam, and Jeffrey W. Herrmann, "Optimizing Urgent Material Delivery by Maximizing Inventory Slack," Technical Report 2012-6, Institute for Systems Research, University of Maryland, College Park, June 12, 2012. Available online at `http://hdl.handle.net/1903/12499.`

Scriabin, Michael, and Roger C. Vergin, “Comparison of Computer Algorithms and Visual Based Methods for Plant Layout,” *Management Science*, Volume 22, Number 2, pages 172-181, 1975.

Simon, Herbert A., 1997a, *Models of Bounded Rationality*, Volume 3, The MIT Press, Cambridge, MA.

Simon, Herbert A., 1997b, *Administrative Behavior*, fourth edition, The Free Press, New York.

Trybus, Thomas W., and Lewis D. Hopkins, “Humans vs. Computer Algorithms for the Plant Layout Problem,” *Management Science*, Volume 26, Number 6, pages 570-574, 1980.

Appendix A

Table A1. Sample standard deviations of the total cost by instance, partition, and separation. (The performance of the all-at-once search is not affected by the partition.)

Instance	Partition	All-at-once	SA1-LA1	SA2-LA1	SA1-LA2	SA2-LA2
SCR12	3333	1,049	3,982	3,646	6,913	4,409
	66		1,219	723	7,154	1,054
SCR15	4443	2,394	6,632	4,418	11,169	6,542
	87		2,654	1,372	8,487	1,896
SCR20	6644	2,677	6,586	4,000	10,080	6,503
	668		4,537	2,613	10,058	5,651
	128		3,549	2,335	8,005	4,349
ELS19	2953	1,922,102	9,189,634	1,263,412	13,009,393	1,231,629
	298		8,354,632	1,406,419	11,259,938	1,053,789
KRA30a	7788	1,629	3,241	3,005	4,588	4,274
	1416		2,229	1,613	3,502	2,901
KRA30b	9696	1,627	2,984	2,881	4,394	4,164
	9912		2,172	2,118	5,124	5,087
	1515		2,238	1,746	3,633	3,250
KRA32	8888	1,698	3,442	3,102	4,770	4,400
	1616		2,147	1,725	3,577	3,062

Appendix B

This appendix describes the greedy heuristic used in the subset assignment procedure SA4. This heuristic groups the departments into N "aggregate departments" based on the flow between them.

Let n be the number of departments and N be the number of aggregate departments (subsets). Then, $r = n / N$ is the number of departments in each aggregate department. The algorithm works by constructing one aggregate department at a time.

1. Set $G_b = \{\}$ for $b = 1, \ldots, N$. Set $H = \{1, \ldots, n\}$.

2. For $b = 1$ to N-1, perform the following steps:

   a. Find departments p and q such that $f_{pq} = \max\left\{ f_{ik} : i,k \in H \right\}$. Add p and q to G_b.

   b. If $r = 2$, then go to step d.

   c. For $h = 3, \ldots, r$, do the following: calculate $\tilde{f}_{bk} = \sum_{i \in G_b} f_{ik}$, find department p such that $\tilde{f}_{bp} = \max\left\{ \tilde{f}_{bk} : k \in H \setminus G_b \right\}$, and add p to G_b.

   d. Remove G_b from H.

3. Set $G_N = H$.