INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD ● INDIA

Research and Publications

The Single Row Facility Layout Problem: State of the Art

Ravi Kothari
Diptesh Ghosh

W.P. No. 2011-12-02
December 2011

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD – 380015
INDIA

The Single Row Facility Layout Problem: State of the Art

Ravi Kothari
Diptesh Ghosh

Abstract

The single row facility layout problem (SRFLP) is a NP-hard problem concerned with the arrangement of facilities of given lengths on a line so as to minimize the weighted sum of the distances between all the pairs of facilities. The SRFLP and its special cases often arise while modeling a large variety of applications. It was actively researched until the mid-nineties. It has again been actively studied since 2005. Interestingly, research on many aspects of this problem is still in the initial stages, and hence the SRFLP is an interesting problem to work on. In this paper, we review the literature on the SRFLP and comment on its relationship with other location problems. We then provide an overview of different formulations of the problem that appear in the literature. We provide exact and heuristic approaches that have been used to solve SRFLPs, and finally point out research gaps and promising directions for future research on this problem.

1 Introduction

In this paper, we study the literature on the Single Row Facility Layout Problem (SRFLP). The problem is known to be NP-hard Beghin-Picavet and Hansen (1982) and has attracted significant attention in recent years. The problem is the following. Consider a set of rectangular facilities in which each of the facilities differ in their lengths. Each facility communicates with every other facility. The cost of communication between a pair of facilities is the product of the transmission intensity between the two facilities and the distance between them. The transmission intensity can be visualized as the number of times the facilities in the pair need to communicate with each other, and the distance between the pair of facilities is measured as the distance between their centroids. So the cost of communication between a pair of facilities is high either if the pair has a high transmission intensity, i.e., they communicate frequently, or if the facilities are located far apart. The total cost of transmission is the sum of the costs of transmission between every pair of facilities. The objective of the SRFLP is to arrange the facilities in a single row along their lengths so that the total cost of transmission is as low as possible. The size of a SRFLP is the number of facilities in the problem. This problem was first proposed by Simmons Simmons (1969). Formally stated the SRFLP is defined as follows:

Given: A set $F = \{1, 2, \ldots, n\}$ of $n > 2$ facilities, where facility j has length l_j, and transmission intensities c_{ij} for each pair (i, j) of facilities, $i, j \in F$, $i \neq j$.

Objective: To find a permutation $\Pi = (\pi_1, \pi_2, \ldots, \pi_n)$ of F that minimizes the expression

$$\sum_{1 \leq i < j \leq n} c_{\pi_i \pi_j} d^{\Pi}_{\pi_i \pi_j}$$

where $d^{\Pi}_{\pi_i \pi_j} = l_{\pi_i}/2 + \sum_{i < k < j} l_{\pi_k} + l_{\pi_j}/2$ is the distance between the centroids of facilities π_i and π_j when the facilities in F are ordered as per the permutation Π.

The SRFLP has been used to model numerous practical situations. It has been a model for arrangement of rooms in hospitals, departments in office buildings or in supermarkets Simmons (1969), arrangement of machines in flexible manufacturing systems Heragu and Kusiak (1988), assignment of files to disk cylinders in computer storage, and design of warehouse layouts Picard and Queyranne (1981). Apart from these direct applications, there are a large number of applications of the special case of the SRFLP in which the facilities have equal lengths. These include assignments of airplanes to gates at an airport terminal Anjos and Yen (2009), triangulation problem of input output tables in economics Laguna et al. (1999), and ranking of teams in sports Martí and Reinelt (2011).

In this paper we survey the literature on the SRFLP. Section 2 places the problem in a hierarchy of related problems. Section 3 presents several ways in which the problem has been formulated in the literature. Sections 4 and 5 present different solution methods that have been used to solve the problem, with the former section dealing with exact methods which guarantee optimal solutions and the latter section dealing with heuristic solution procedures. Finally, Section 6 points out gaps in the literature and suggests promising directions for future research on this problem.

2   Relations to other problems

The SRFLP can be placed in a hierarchy of problems dealt with in the literature, some of whose members are special cases of other members. We present this hierarchy below, starting from the most general problem in the hierarchy and ending with the most specialized problem.

Space allocation problem: The objective in a space allocation problem (SAP) is to shape and orient facilities on a floor plan so as to achieve a given efficiency measure. Such a problem arises when an architect is trying to arrange facilities of equal area but unspecified shape so as to minimize a given linear combination of distances between all pairs of facilities (see Simmons (1969)). This is not strictly a facility location problem, because in a facility location problem there are pre-defined set of locations which are separated by a fixed distance and it is only required to determine which facilities are to be located at which locations. In the SAP the locations are not defined and it is up to the architect to design the entire floor plan.

Single row facility layout problem: The single row facility layout problem (SRFLP) is a special case of the SAP where facilities have fixed lengths and are to be arranged in a linear fashion. It is an ordering problem with facilities of non-uniform length being arranged along a single row (see, e.g., Anjos et al. (2005)). The SRFLP is also known as the one dimensional space allocation problem (ODSAP).

Generalized linear ordering problem: An interesting location problem that is closely related to SRFLP is the generalized linear ordering problem (GLOP). Here one is given n facilities, n locations along a line, and the transmission intensity between each pair of facilities, and one needs to find the one-one assignment of n facilities to the n locations so as to minimize the total transmission cost (see, e.g., Picard and Queyranne (1981)).

Linear ordering problem: The linear ordering problem (LOP) is a special case of both the SRFLP and the GLOP. It can be described as a SRFLP in which all facilities have the same length. It can also be seen as a GLOP with regularly placed locations separated by the common length of the facilities. The LOP is also referred to as the linear arrangement problem in the literature (see, e.g., Picard and Queyranne (1981)).

3   Formulations of SRFLP

We now describe the different integer and mixed integer programming formulations of the SRFLP that are available in the literature. In this section, we describe these formulations in chronological order. For consistency, we use the same notation as used in the formal description of the SRFLP in Section 1.

Note that the term c_{ij} has been used in different connotations in the literature. It has been defined as cost by Simmons Simmons (1969), frequency of travel by Heragu and Kusiak Heragu and Kusiak (1988), affinity by Romero and Sánchez-Flores Romero and Sánchez-Flores (1990), and transition probabilities by Picard and Queyranne Picard and Queyranne (1981).
The first formulation of the problem which was proposed by Simmons Simmons (1969) is the following.

$$\text{Minimize} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} d_{ij}.$$

The paper showed that this formulation is equivalent to

$$\text{Minimize} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} b_{ij},$$

where b_{ij} is sum of the lengths of all facilities between facility i and facility j in a permutation. The formulation used $n(n-1)/4$ variables. The constraints were not explicitly formulated in the paper.

Love and Wong Love and Wong (1976) provided the first linear mixed integer programming formulation for the SRFLP. The formulation was as follows.

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij}(R_{ij} + L_{ij}) \\
\text{Subject to} \quad & R_{ij} - L_{ij} = x_i - x_j + \frac{1}{2}(l_j - l_i) \\
& x_i - x_j + M(\alpha_{ij}) \geq l_i \\
& x_j - x_i + M(1 - \alpha_{ij}) \geq l_j \\
& l_i \leq x_i \leq \sum_{i=1}^{n} l_i \\
& \alpha_{ij} = 0, 1 \\
& R_{ij}, L_{ij}, x_1, ..., x_n \geq 0,
\end{aligned}
$$

where x_i is the endpoint of facility i farthest from the origin; α_{ij} is 1 if facility i is to the left of facility j, and 0 otherwise; R_{ij} is the distance between the centroid of facility i and centroid of facility j if $\alpha_{ij} = 0$, and 0 otherwise; L_{ij} is the distance between centroid of facility i and centroid of facility j if $\alpha_{ij} = 1$ and 0 otherwise; and M is an arbitrarily large number.

This formulation used $n(n-1)/2$ binary variables and had $3n(n-1)/2$ constraints excluding the non-negativity constraints and lower and upper bounds on x_i.

Picard and Queyranne Picard and Queyranne (1981) proposed the the same formulation as by Simmons Simmons (1969). However, their paper was the first to explicitly state the idea that every solution to the SRFLP is in fact a permutation of the set of facilities.

Ravi Kumar et al. Ravi Kumar et al. (1995) presented a quadratic assignment problem formulation for the GLOP, which is a special case of the SRFLP in which a number of distinct facilities are to be assigned to an equal number of equidistant locations which lie on a straight line. The mathematical formulation was the following.

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{k'=1}^{n} x_{ik}x_{jk'}c_{ij}|k-k'|$$

$$\text{Subject to} \quad \sum_{i=1}^{n} x_{ik} = 1, \quad k = 1, 2, ..., n$$

$$\sum_{k=1}^{n} x_{ik} = 1, \quad i = 1, 2, ..., n$$

$$x_{ik} \in 0,1 \quad i, k = 1, 2, ..., n,$$

where x_{ik} is 1 if facility i is assigned to location k, and 0 otherwise. This formulation used n^2 binary variables and had $2n$ constraints.

Amaral Amaral (2006) introduced a new mixed integer linear programming model for the SRFLP. This formulation used additional variables of the form α_{ij} which assume a value of 1 if facility i is to the left of facility j, and 0 otherwise. It also used two polytopes, $H_n = \{\alpha \in \Re_{+}^{n(n-1)/2} : 0 \leq \alpha_{ij} + \alpha_{jk} - \alpha_{ik} \leq 1, 1 \leq i < j < k \leq n\}$ and $D_n = \{d \in \Re^{n(n-1)/2} : d_{ij} \geq (l_i + l_j)/2, 1 \leq i < j \leq n\}$. The formulation was as follows.

$$\text{Minimize} \quad \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij}d_{ij}$$

$$\text{Subject to} \quad d_{ij} \geq \sum_{k<i} l_k\alpha_{ki} + \sum_{k>i} l_k(1-\alpha_{ik}) - \sum_{k<j} l_k\alpha_{kj} -$$

$$\sum_{k>j} l_k(1-\alpha_{kj}) + (l_i - l_j)/2 \quad 1 \leq i < j \leq n$$

$$d_{ij} \geq \sum_{k<i} l_k\alpha_{ki} - \sum_{k>i} l_k(1-\alpha_{ik}) + \sum_{k<j} l_k\alpha_{kj} +$$

$$\sum_{k>j} l_k(1-\alpha_{kj}) + (l_j - l_i)/2 \quad 1 \leq i < j \leq n$$

$$\alpha \in H_n$$

$$d \in D_n$$

$$\alpha_{ij} \in 0,1 \quad 1 \leq i < j \leq n.$$

The formulation used $n(n-1)/2$ binary variables and $n(n-1)/2$ continuous variables. There were $n(n-1)(2n-1)/3$ constraints excluding the non-negativity constraints and the bounds on the variables.

Amaral Amaral (2009) proposed a new "betweenness" based formulation for the SRFLP. The formulation defined a set U_n as a convex hull of all vectors $\zeta = (\zeta_{ijk})$ where ζ_{ijk} is 1 when facility k lies between facilities i and j, and 0 otherwise. The formulation was the following.

$$\text{Minimize}_{\zeta \in U_n} \sum_{i<j, k<j} (c_{ij} l_k - c_{ik} l_j) \zeta_{ijk} + \sum_{i<j} \left(\frac{c_{ij}}{2} (l_i + l_j) + \sum_{k>j} c_{ij} l_k \right)$$
$$\text{Subject to} \quad \zeta_{ijk} + \zeta_{ikj} + \zeta_{jki} = 1, \quad i < j < k$$
$$\zeta \in \{0, 1\}^{n(n-1)(n-2)/2}$$

It used $n(n-1)(n-2)/2$ binary variables and had $\binom{n}{3}$ constraints.

4 Exact solution methods

Exact solution methods for the SRFLP are computationally expensive and have high memory requirements. In this section, we describe some of the exact methods reported in the literature to solve the SRFLP.

Branch and Bound

Branch and bound techniques are computationally very expensive when applied to SRFLP of very small sizes, hence not much research has been done in obtaining an exact branch and bound procedure for SRFLP of sizes larger than 12. Simmons Simmons (1969) proposed the first branch and bound algorithm to find an exact solution to the SRFLP. This algorithm was characterized by a stepwise build-up of the feasible ordering of the facilities i.e., the permutation of facilities was not built up until the late stages of the algorithm. For this very reason the technique is also known as the *branch and build up* approach. SRFLP instances of size up to 11 were solved using this technique.

Mathematical programming

Love and Wong Love and Wong (1976) formulated the SRFLP as a linear mixed integer program and solved it using the IBM MIP code. The code was capable of solving SRFLP instances with 10 facilities or less. For larger instances, several runs were needed to to obtain the solution, along with a heuristic judgement on fixing the positions of some facilities. However, improved results were obtained when SRFLP instances with 20 to 30 facilities were solved using the nonlinear model of Heragu and Kusiak Heragu and Kusiak (1991). The nonlinear formulation was solved by transforming the constrained nonlinear model into an unconstrained model using a penalty method. A new mixed integer linear programming formulation of the SRFLP was proposed by Amaral Amaral (2006). The presence of facet defining inequalities made the formulation stronger than that of Love and Wong Love and Wong (1976). SRFLP instances with up to 15 facilities from Simmons (1969), Love and Wong (1976), Heragu and Kusiak (1991) were solved more efficiently than in previous studies reported in the literature. Another mixed integer linear programming formulation was proposed by Amaral Amaral (2008). This formulation was obtained after a series of mathematical transformations on a 0-1 quadratic programming model presented in the same paper. Codes based on this formulation are more efficient than those based on previously published mixed integer formulations. This code could solve SRFLP instances with sizes up to 18 that were proposed in Simmons (1969), Love and Wong (1976), Heragu and Kusiak (1991) within reasonable time.

Dynamic programming

Picard and Queyranne Picard and Queyranne (1981) reported the first dynamic programming algorithm to the solve the SRFLP. Although the algorithm was however successfully applied to the SRFLP instances with up to 15 facilities, the high memory requirements of this technique (it requires $O(n\ 2^n)$ time) has rendered it unattractive for the research community. Another dynamic programming algorithm was presented by Kouvelis and Chiang Kouvelis and Chiang (1996). It was able to solve SRFLP instances with 20 facilities.

Cutting planes approach

Amaral Amaral (2009) suggested a linear programming based cutting plane approach using betweenness variables. The paper presented a partial linear description of a polytope based on the betweenness variables and obtained a new lower bound for the SRFLP by optimizing the linear program (LP) over the partial description augmented with cutting planes when necessary. The LP formulation used in this paper is given in Section 3 and it was found that the lower bound obtained using this cutting plane approach equalled the cost of an optimal solution in each of the SRFLP instances with up to 35 facilities on which it was tested. The approach in this paper produced results identical to those using the semidefinite programming relaxation approach of Anjos and Vannelli Anjos and Vannelli (2008), but required less time.

Branch and Cut

Codes based on the work by Amaral Amaral (2006; 2008) suffered from weak lower bounds and had high computation times and memory requirements. The first in-depth polyhedral study of the SRFLP by Amaral and Letchford Amaral and Letchford (2011) achieved significant progress in this direction. It derived valid inequalities and facets for the SRFLP and proposed a branch and cut algorithm based on the distance variable d_{ij} to solve the SRFLP. A branch and cut algorithm uses a combination of a branching rules and cutting planes to reduce the feasible region of a mathematical programming problem. Amaral and Letchford used a cheap initial LP relaxation with only $O(n^2)$ non-zero coefficients and applied exact separation routines for triangle and special strengthened pure negative type inequalities and heuristic ones for clique, rounded positive semi definite and star inequalities. They suggested a specialised branching rule to avoid the use of additional binary variables and used a primal heuristic based on multi-dimensional scaling to obtain feasible permutations for the SRFLP. The approach solved SRFLP instances of size up to 30 taken from Simmons (1969), Heragu and Kusiak (1988), Anjos and Vannelli (2008) within reasonable computing times. The algorithm also provided tight lower and upper bounds at the root node. When the algorithm was applied to three SRFLP instances with 110 facilities, it was found that the percentage gap between the upper bound and the lower bound varied from 3% to 4% when a run time limit of 2.5 days was set.

Semidefinite programming

Semidefinite programming (SDP) refers to the class of optimization problems where a linear function of a matrix variable X is optimized subject to linear constraints on the elements of X and an additional constraint that X be positive semi definite. Linear programming problems are a special case, namely when all the matrices involved are diagonal (see Anjos et al. (2005)). Anjos et al. Anjos et al. (2005) presented a SDP relaxation of the SRFLP, whose solution provided a lower bound on the optimal value of the SRFLP. They also provided a heuristic to extract the permutation of facilities from the optimal matrix solution of the SDP relaxation. The algorithm was applied on SRFLP instances of size upto 30 taken from Ravi Kumar et al. (1995), Heragu and Alfa (1992) and it was

found that the permutations obtained were competitive with the best available in the literature. The SDP approach was also applied to randomly generated instances with up to 80 facilities inAnjos et al. (2005) and their lower bounds to the costs of optimal solutions for the instances were obtained.

The SDP relaxation presented by Anjos et al. Anjos et al. (2005) was further tightened by Anjos and Vannelli Anjos and Vannelli (2008) using triangle inequalities as cutting planes. Problem instances of size upto 30 were taken from Nugent et al. (1968), Simmons (1969), Heragu and Kusiak (1988), Ravi Kumar et al. (1995) and their global optimal solutions were obtained. Some of these had remained unsolved since 1988. The algorithm was capable of solving problem instances with up to 25 facilities within a few hours and upto 30 facilities "in several dozen hours" [sic].

The SDP relaxation presented by Anjos et al. Anjos et al. (2005) had $O(n^3)$ linear constraints and it required large models to be solved. In order to overcome the difficulty of large computation time requirements, an alternate SDP relaxation with $O(n^2)$ linear constraints was presented by Anjos and Yen Anjos and Yen (2009). This alternate SDP relaxation could handle problems with up to 100 facilities at the cost of a marginal deterioration of the solutions obtained by Anjos et al. Anjos et al. (2005). The CPU times required using the formulation by Anjos and Yen Anjos and Yen (2009) was found to be significantly lower than that obtained using the formulation by Anjos et al. Anjos et al. (2005) for the same SRFLP instances.

A recent study by Hungerländer and Rendl Hungerländer and Rendl (2011b) further improved the tightness of the SDP relaxations of Anjos et al. Anjos et al. (2005) and Anjos and Yen Anjos and Yen (2009). It used a suitable combination of optimization methods to deal with stronger but more expensive relaxations. The method by Hungerländer and Rendl Hungerländer and Rendl (2011b) built on the subgradient optimization technique and dealt with triangle inequality cuts and the LS cuts through Langrangian duality. Hungerländer and Rendl Hungerländer and Rendl (2011a) reported results from applying the above SDP approach to SRFLP with upto 100 facilities and obtained optimal solutions for several SRFLP instances from the literature with up to 42 facilities. For problems with upto 20 instances the integer linear programming approaches of Amaral Amaral (2009) and Amaral and Letchford Amaral and Letchford (2011) were preferable to SDP approaches while the SDP approach of Hungerländer and Rendl Hungerländer and Rendl (2011a) outperformed the other approaches for large instances. When compared with the study of Anjos and Yen Anjos and Yen (2009), the study of Hungerländer and Rendl Hungerländer and Rendl (2011a) reduced the known gaps between the lower bounds obtained and the best permutations for the SRFLP instances with 60 to 100 facilities.

5   Heuristic solution methods

Since exact algorithms for the SRFLP are computationally expensive, they have been applied to relatively small instances, with up to 42 facilities. Researchers use heuristics to solve larger sized SRFLP instances. These are much faster than exact algorithms, but do not guarantee optimal solutions. In this section, we review heuristics for the SRFLP under two heads, construction heuristics and improvement heuristics. Note that most improvement heuristics are known in the literature as metaheuristics, since they only provide a framework for solving combinatorial problems in general, and have to be tailored appropriately to solve specific classes of problems.

Construction heuristics

A construction heuristic when applied to a SRFLP progressively constructs the sequence of facilities until the complete permutation is obtained. Usually the position of a single facility is fixed in every iteration of the heuristic. However in some construction heuristics for the SRFLP, multiple facilities are assigned to locations in a single iteration of the heuristic.

Heragu and Kusiak Heragu and Kusiak (1988) presented the first construction heuristic for the SRFLP. In each iteration, the heuristic modified an adjusted flow matrix which was computed based on the transmission matrix and the solution built thus far, selected a pair of facilities, and connected them to form a part of the solution. This heuristic output optimal solutions for SRFLP instances with size up to four. It was applied to nine instances of sizes up to 20 from Nugent et al. (1968) and optimal solutions was found for six of these instances.

Ravi Kumar et al. Ravi Kumar et al. (1995) presented another greedy heuristic. This heuristic ignored the lengths of the facilities, and tried to assign facilities with the largest inter-facility transmission intensity to adjacent locations in the solution. It differed from other heuristic in the literature since it allowed the assignment of more than one facilities to an existing sequence of facilities at any iteration in the heuristic. It was tested on problem instances with up to 30 facilities from Nugent et al. (1968), Simmons (1969), Love and Wong (1976), Heragu and Kusiak (1988; 1991) and it obtained better solutions than what were known at that time in significantly lesser time.

A third greedy construction heuristic, derived from the NEH heuristic (developed for a flow shop scheduling problem), was proposed by Braglia Braglia (1997). The heuristic was a two step procedure in which the first step selected one facility and the second step selected the best location for that facility in the solution. It was applied to randomly generated test problems with up to 30 facilities and the paper concluded that the study of the performance of a heuristic should focus on the class of problems that not only consider the number of facilities but also the density of the matrix of transmission intensities.

Djellab and Gourgand Djellab and Gourgand (2001) proposed an insertion based two step construction heuristic. The heuristic was applied to problem instances with upto 30 facilities from Nugent et al. (1968), Heragu and Kusiak (1991). It was found that except for one instance it either output the best known solution or outperformed all other methods available in the literature until 2001.

Improvement heuristics

Improvement heuristics start with one or more permutations of facilities and iteratively improve a starting solution(s) of facilities until the solution cannot be improved further or a stopping criterion is reached. The classification of improvement heuristics is based on the whether they work on improving a single current solution at each iteration (single solution based heuristics) or use more than one solutions and combine them together to generate new solutions at each iteration (population based heuristics). In general, the solutions obtained using improvement heuristics do not guarantee optimality, or even a bound on solution quality. In case of the SRFLP, heuristics such as tabu search and simulated annealing, which are commonly implemented as single solution based heuristics have also been implemented as population based heuristics, with the difference that in each iteration, only one of the permutations was chosen and changed by the heuristic. The following are the improvement heuristics suggested in the literature to solve the SRFLP.

Tabu search

Samarghandi and Eshghi Samarghandi and Eshghi (2010) used tabu search to solve the SRFLP. Their tabu search implementation was a population based variant of the usual tabu search. In order to generate initial solutions, Samarghandi and Eshghi used a special case of SRFLP in which the transmission intensities between departments were assumed to be equal. Based on this, their algorithm created a set of initial solutions and stored them as a set of current solutions. At each iteration of the algorithm, a probabilistic technique was used to select one solution from the set of current solutions and the algorithm searched its 2-opt neighborhood. If the new permutation improved the objective function of the ABSMODEL described in Heragu and Kusiak (1991), then it

was added to the set of current solutions and the tabu list was updated with the current exchange for a user-specified fixed number of iterations. This meant that these facilities would not be exchanged for the specified number of iterations unless such an exchange was the only one that improved the objective of the ABSMODEL, or the solution generated by this exchange had the best objective function obtained thus far by the search process. The number of solutions in the set of current solutions was restored by deleting the worst solution in the set after adding the new solution. These iterations continued until no further improvement was achieved for a user-specified number of iterations. Additionally, the algorithm employed ideas of diversification and intensification during the search process. On reaching the stopping criterion the algorithm chose the best solution from set of current solutions. It then performed a final intensification step on this solution and output the result of the intensification step.

This algorithm was applied to 30 SRFLP instances of sizes up to 30 from Solimanpur et al. (2005), Amaral (2006), Anjos and Vannelli (2008) and it output optimal solutions for all the problem instances within reasonable time. In 20 SRFLP instances with sizes up to 80 from Anjos et al. (2005), the algorithm improved upper bounds on the costs of optimal solutions known at that time.

Simulated annealing

Romero and Sánchez-Flores Romero and Sánchez-Flores (1990) first applied simulated annealing to the SRFLP. They compared the performance of several simulated annealing algorithm implementations in which various combinations of elementary exchange and insertion strategies were studied to find the neighbor of a solution, and reported that insertion based neighborhoods are better than exchange based neighborhoods. They also developed four broad methods, two based on simulated annealing and two based on local improvement methods and created 70 variations of the algorithms to solve the SRFLP. The final solutions obtained by all the variants were improved using a 3-opt local improvement method called 3-LOC. Romero and Sánchez-Flores concluded that the local improvement methods are better suited for small sized problems, while simulated annealing works better for instances with sizes around 40 or 50.

Kouvelis and Chiang Kouvelis and Chiang (1992) reported another simulated annealing based approach in the context of a FMS application. They used a 2-opt neighborhood structure. Their algorithm was tested on small sized SRFLP instances, and the paper concluded that if the parameters for simulated annealing are finely tuned and good permutations are provided as initial seeds then high quality solutions are output using simulated annealing.

Heragu and Alfa Heragu and Alfa (1992) reported an experimental analysis of five different algorithms. They compared 2-opt, 3-opt, a modified penalty algorithm, simulated annealing, and hybrid simulated annealing (HSA) algorithm. In a HSA, the initial solution was generated by the modified penalty method and then simulated annealing was applied on the initial solution. The modified penalty algorithm was same as the one used by Heragu and Kusiak Heragu and Kusiak (1991), but augmented by a final improvement step using 2-opt. HSA was applied on the SRFLP instances from Heragu and Kusiak (1991) and it was observed that HSA output optimal solutions for small sized instances and improved best known solutions for instances with sizes of 20 and 30. In general the run time of HSA was found to be higher than that of simulated annealing.

Ant colony optimization

Solimanpur et al. Solimanpur et al. (2005) proposed an ant algorithm for the SRFLP. In each iteration of their algorithm a fresh set of ants were created. Each ant was an agent that chose moves. A move assigned a facility to a location. For every move, an ant had a list that kept track of the facilities which had already been assigned. All the moves by an ant were stored in the ant's memory. This memory was used at the end of an iteration to update trail levels, which are analogous

to pheromone deposits. Moves were performed through a heuristic procedure. A factor known as the desirability of a move was computed for every move. The trail level for each move indicated how frequently that particular move had been selected by other ants in the algorithm. The desirability and the trail level of the moves were combined to compute the probability of selecting a particular move by an ant. The solution generated by each ant was improved locally using a 2-opt structure. After the improvement the trail levels were updated, the ants were destroyed, and the next iteration began. The total number of iterations was set by the user to control the run time of the algorithm. The number of ants in each iteration was also specified by the user.

Naïve implementations of this algorithm require $O(n^4)$ time and are thus computationally expensive. Solimanpur et al. suggested an effective way of implementing the algorithm in a $O(n^3)$ time. They applied the algorithm on SRFLP instances with up to 30 facilities obtained from Nugent et al. (1968), Simmons (1969), Love and Wong (1976), Beghin-Picavet and Hansen (1982) and observed that the proposed algorithm obtained the best solutions for all the instances. It beats the solutions reported by Heragu and Kusiak Heragu and Kusiak (1988) and Ravi Kumar et al. Ravi Kumar et al. (1995).

Scatter search

Kumar et al. Kumar et al. (2008) applied scatter search to the SRFLP. Their diversification generation and the subset generation methods were adopted from Glover (1998). The reference set had an equal number of good solutions and diverse solutions and the left and right insertion heuristic was used as an improvement method. They used the algorithm to solve SRFLP instances from Solimanpur et al. (2005) and obtained results that were identical to those in Solimanpur et al. (2005), except when the problem size was 30. In these cases, scatter search output a result which was better than that obtained by any other study in the literature until 2008.

Particle swarm optimization

Samarghandi et al. Samarghandi et al. (2010) applied particle swarm optimization to the SRFLP. They used a factoradic encoding/decoding to map the discrete feasible space of the SRFLP to a continuous space. The factoradic number system (see, e.g., Knuth (1997)) was used to establish a one-one map between the set of all permutations of the facilities in SRFLP and the set of all factoradic numbers and hence a decimal number corresponding to each permutation was obtained. Thus it associated a unique number with every permutation in the particle swarm optimization algorithm They then used the inverse mapping to retreive the required permutation corresponding to the best selection. At the beginning of the algorithm, particles which, in their implementation are numbers which correspond to permutations, were generated randomly from the interval $[0, n! - 1]$ to ensure diversification in the search process. Velocities of the initial randomly generated particles were also generated randomly in an appropriate interval so as to ensure that the particles do not go out of the feasible region. At each iteration, new positions of the particles were computed based on their velocities, and the particles were moved to new positions. Costs were computed for all particles at their new positions. The current solution and the best solution obtained so far were updated. Every particle in the algorithm underwent an intensification process after a fixed number of iterations. The intensification process was carried out by applying local search using a 2-opt neighborhood structure. The algorithm terminated after an user specified termination condition was met, and the best solution encountered by the algorithm was output.

Samarghandi et al. applied the algorithm to the problem sets used in Solimanpur et al. (2005) and obtained the same results as reported in Solimanpur et al. (2005). They also used the algorithm to solve SRFLP instances from Amaral (2006), Anjos and Vannelli (2008) with up to 30 facilities on which heuristics had never been applied before. The algorithm obtained optimal solutions for eight of the 12 problem instances.

Genetic algorithm

The only application of genetic algorithms to solve the SRFLP was by Datta et al. Datta et al. (2011). In the algorithm, solutions in the population were represented as a permutation of the facilities. The initial population was initialized in four ways to ensure diversification in the population. The selection operator used the binary tournament selection to form the mating pool. After the creation of the mating pool, the crossover operator selected two parent solutions from the mating pool. A new (child) solution was generated as follows. Initially none of the facilities in the child were allocated to particular locations. One parent was chosen at random. A facility was chosen at random from this parent. The algorithm tried to assign that facility in the child in the same location as that in the chosen parent. If the location was already assigned to some other facility, then this facility was kept in a reserve pool. Once all the facilities not in the reserve pool had been assigned locations in the child, the facilities in the reserve pool were assigned randomly to the unassigned locations. This crossover operation was used to generate the same number of children as the number of solutions in the mating pool. Every child solution was then modified using a mutation operator. The two sets of solutions, i.e., the mating pool and the set of child solutions were merged, and an elite preserving operator was applied to pick the top 50% of these solutions from this pool to form the next generation. The algorithm terminated after a user-specified number of generations were produced, and the best solution encountered by the algorithm was output.

Dutta et al. used the genetic algorithm to solve 14 problem instances from Amaral (2009), Anjos et al. (2005) with size up to 30 facilities whose optimal solutions were known, and 20 instances from Anjos et al. (2005) of size ranging from 60 to 80 whose optimal solutions were not known. The first 14 problems were solved to optimality and the algorithm improved the best known costs for nine of next 20 instances.

6 Summary of past research and future directions

Although the SRFLP is a problem that has been known in the literature for a long time, research on most aspects of the problem has not been thorough. In this section, we summarize the broad nature of past research that we have observed in the SRFLP literature, and point out research directions for this problem.

The main techniques used for obtaining optimal solutions to a combinatorial optimization problem are enumeration based algorithms like branch and bound algorithms, cutting plane algorithms, and dynamic programming. In the SRFLP literature, the initial focus seemed to have been on enumeration based algorithms. However, this line of research has not continued up to recent times, with the last branch and bound algorithm being proposed in 1969 (see Simmons (1969)). A problem with these approaches seems to be that good bounds have not been obtained for the problem, which implies that time and memory requirements for such algorithms are very high. Developing such bounds is an interesting line of study for SRFLPs. In addition, new branching rules can be considered and evaluated. Very recently, branch and cut methods have been studied for a semidefinite programming based relaxations of this problem (see Hungerländer and Rendl (2011a)). This paper itself points out to classes of cuts that have not been studied for the relaxation. Cutting planes have been examined quite thoroughly in the work presented in Anjos and Vannelli (2008), Amaral and Letchford (2011), Hungerländer and Rendl (2011a). As usual, further work in this approach is to strengthen existing cuts and to devise new families of cuts, which can then either be used in cutting plane algorithms, or be incorporated in branch and cut algorithms. Dynamic programming, which often results in good algorithms for many other problem classes had been dealt with in only one paper Picard and Queyranne (1981). However, the approach was seen to be extremely computation intensive, and there has not been any further advances using this technique.

Exact algorithms have not been able to generate optimal solutions to SRFLP instances with more than 42 facilities (see Hungerländer and Rendl (2011a)). For larger instances, research has focused on heuristics. We now examine the achievements and future research directions in the field of heuristics for the SRFLP. We first look at construction heuristics and then at improvement heuristics.

Till date, only four construction heuristics have been reported in the literature. All of these are greedy heuristics, among which one (see Ravi Kumar et al. (1995)) does not consider the lengths of facilities. Sophisticated construction heuristics and polynomial time approximation schemes for the SRFLP have not appeared in the literature, and it is interesting to design such heuristics. Interestingly, even for the greedy heuristics reported in the literature, the performance of the heuristics have only been compared through experiments; no worst-case or average case performance analysis or dominance analysis has been carried out for these heuristics. In many problems, analysis of the performance of construction heuristics often leads to polynomially solvable special cases of NP-Hard problems. Since the construction heuristics for the SRFLP have not been analysed mathematically, SRFLP does not yet have classes of instances that are proven to be polynomially solvable.

Improvement heuristics have been implemented for the SRFLP and tested on large sized instances only since 2005. Hence there is a limited number of studies on the application of these heuristics. The performance of many important ones like variable neighborhood search, path relinking, and GRASP have not been tested for the SRFLP. Even in the case of most heuristics that have been tested, there is only one study testing a particular heuristic. The studies have also mostly restricted themselves to the more elementary features of the heuristics. For example, in the studies that used local search, the most common neighborhood structure encountered was also the most elementary, viz. the 2-opt neighborhood structure. More elaborate and more effective neighborhoods like 3-opt neighborhoods and exponential neighborhoods have never been used. This is possibly due to the fact that naïve implementation of neighborhoods are quite expensive for SRFLP as compared to other popular combinatorial optimization problems like the traveling salesman problem or simple plant location problems. Similarly, more advanced techniques in tabu search, and advanced operators in genetic algorithms have also never been studied in the context of the SRFLP.

An interesting area of research in combinatorial problems which has so far been overlooked in the case of SRFLP is that of post-optimality analysis. Given a problem and an optimal solution to the problem, post-optimality analysis for combinatorial optimization problems answers questions about the robustness of the optimal solution. In other words, it finds out the ranges of values within which problem parameters should lie for the given optimal solution to remain optimal. It also looks at the variation of the objective function value of a problem when a problem parameter is varied. Such problems are important from a practical standpoint. In practice, the data on which a problem is modeled undergoes changes quite frequently. If the model is difficult to solve, it is difficult to re-optimize the problem repeatedly when such data changes occur. Results from post-optimality analysis can point out when an optimal solution at hand is still optimal for the changed data, and the decision maker does not need to re-optimize the model. In case the problem is a so-called design problem, where re-optimization is not an option, as in many applications of the SRFLP, the analysis tells a decision maker when the current permutation of facilities is the best that can be achieved. Such analysis is quite common (see, e.g., Greenberg (1998) for a detailed review). An interesting direction of research is to obtain analytical expressions for the ranges of parameters and describe algorithms to obtain these bounds for large sized SRFLPs.

References

Amaral, A. and Letchford, A. N. (2011). A polyhedral approach to the single row facility layout problem. Available at `http://eprints.lancs.ac.uk/id/eprint/49043`.

Amaral, A. R. S. (2006). On the exact solution of a facility layout problem. *European Journal of Operational Research*, 173(2):508–518.

Amaral, A. R. S. (2008). An Exact Approach to the One-Dimensional Facility Layout Problem. *Operations Research*, 56(4):1026–1033.

Amaral, A. R. S. (2009). A new lower bound for the single row facility layout problem. *Discrete Applied Mathematics*, 157(1):183–190.

Anjos, M., Kennings, a., and Vannelli, a. (2005). A semidefinite optimization approach for the single-row layout problem with unequal dimensions. *Discrete Optimization*, 2(2):113–122.

Anjos, M. F. and Vannelli, A. (2008). Computing Globally Optimal Solutions for Single-Row Layout Problems Using Semidefinite Programming and Cutting Planes. *INFORMS Journal on Computing*, 20(4):611–617.

Anjos, M. F. and Yen, G. (2009). Provably near-optimal solutions for very large single-row facility layout problems. *Optimization Methods and Software*, 24(4-5):805–817.

Beghin-Picavet, M. and Hansen, P. (1982). Deux problèmes daffectation non linéaires. *RAIRO, Recherche Opérationnelle*, 16(3):263–276.

Braglia, M. (1997). Heuristics for single-row layout problems in flexible manufacturing systems. *Production Planning & Control*, 8(6):558–567.

Datta, D., Amaral, A. R., and Figueira, J. R. (2011). Single row facility layout problem using a permutation-based genetic algorithm. *European Journal of Operational Research*, 213(2):388–394.

Djellab, H. and Gourgand, M. (2001). A new heuristic procedure for the single -row facility layout problem. *International Journal of Computer Integrated Manufacturing*, 14(3):270–280.

Glover, F. (1998). A template for scatter search and path relinking. *Lecture notes in computer science*, 1363:13–54.

Greenberg, H. (1998). *Advances in computational and stochastic optimization, logic programming, and heuristic search*, chapter An annotated bibliography for post- solution analysis in mixed integer programming and combinatorial optimization, pages 97–148. Kluwer Academic Publishers, Boston, MA, USA.

Heragu, S. S. and Alfa, A. S. (1992). Experimental analysis of simulated annealing based algorithms for the layout problem. *European Journal of Operational Research*, 57(2):190–202.

Heragu, S. S. and Kusiak, A. (1988). Machine Layout Problem in Flexible Manufacturing Systems. *Operations Research*, 36(2):258–268.

Heragu, S. S. and Kusiak, A. (1991). Efficient models for the facility layout problem. *European Journal Of Operational Research*, 53:1–13.

Hungerländer, P. and Rendl, F. (2011a). A computational study for the single-row facility layout problem. Available at `www.optimization-online.org/DB\_FILE/2011/05/3029.pdf`.

Hungerländer, P. and Rendl, F. (2011b). Semidefinite relaxations of ordering problems. *Mathematical Programming B*.

Knuth, D. (1997). *Seminumerical algorithms (3rd ed.). The art of computer programming (Vol. 2).* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc..

Kouvelis, P. and Chiang, W.-C. (1992). A simulated annealing procedure for single row layout problems in flexible manufacturing systems. *International Journal of Production Research*, 30(4):717–732.

Kouvelis, P. and Chiang, W.-C. (1996). Optimal and Heuristic Procedures for Row Layout Problems in Automated Manufacturing Systems. *Journal of the Operational Research Society*, 47(6):803–816.

Kumar, S., Asokan, P., Kumanan, S., and Varma, B. (2008). Scatter search algorithm for single row layout problem in fms. *Advances in Production Engineering & Management*, 3(4):193–204.

Laguna, M., Martí, R., and Campos, V. (1999). Intensification and diversification with elite tabu search solutions for the linear ordering problem. *Computers & OR*, 26(12):1217–1230.

Love, R. F. and Wong, J. Y. (1976). On solving a one-dimensional space allocation problem with integer programming. *INFOR*, 14(2):139–144.

Martí, R. and Reinelt, G. (2011). *The Linear Ordering Problem.* Springer-Verlag Berlin Heidelberg.

Nugent, C. E., Vollman, T. E., and Ruml, J. (1968). An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research*, 16(1):150–173.

Picard, J.-C. and Queyranne, M. (1981). On the one-dimensional space allocation problem. *Operations Research*, 29(2):371–391.

Ravi Kumar, K., Hadejinicola, G. C., and Lin, T.-L. (1995). A heuristic procedure for the single-row facility layout problem. *European Journal of Operational Research*, 87(1):65–73.

Romero, D. and Sánchez-Flores, A. (1990). Methods for the one-dimensional space allocation problem. *Computers & Operations Research*, 17(5):465–473.

Samarghandi, H. and Eshghi, K. (2010). An efficient tabu algorithm for the single row facility layout problem. *European Journal of Operational Research*, 205(1):98–105.

Samarghandi, H., Taabayan, P., and Jahantigh, F. F. (2010). A particle swarm optimization for the single row facility layout problem. *Computers & Industrial Engineering*, 58(4):529–534.

Simmons, D. M. (1969). One-Dimensional Space Allocation: An Ordering Algorithm. *Operations Research*, 17(5):812–826.

Solimanpur, M., Vrat, P., and Shanker, R. (2005). An ant algorithm for the single row layout problem in flexible manufacturing systems. *Computers & Operations Research*, 32(3):583–598.