European Journal of Operational Research 246 (2015) 800–814

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Fast local search for single row facility layout

Gintaras Palubeckis*

Faculty of Informatics, Kaunas University of Technology, Studentu 50-408, 51368 Kaunas, Lithuania

ARTICLE INFO

Article history:
Received 11 April 2014
Accepted 20 May 2015
Available online 30 May 2015

Keywords:
Combinatorial optimization
Single row facility layout
Local search
Variable neighborhood search

ABSTRACT

Given n facilities of prescribed lengths and a flow matrix, the single row facility layout problem (SRFLP) is to arrange the facilities along a straight line so as to minimize the total arrangement cost, which is the sum of the products of the flows and center-to-center distances between facilities. We propose interchange and insertion neighborhood exploration (NE) procedures with time complexity $O(n^2)$, which is an improvement over $O(n^3)$-time NE procedures from the literature. Numerical results show that, for large SRFLP instances, our insertion-based local search (LS) algorithm is two orders of magnitude faster than the best existing LS techniques. As a case study, we embed this LS algorithm into the variable neighborhood search (VNS) framework. We report computational results for SRFLP instances of size up to 300 facilities. They indicate that our VNS implementation offers markedly better performance than the variant of VNS that uses a recently proposed $O(n^3)$-time insertion-based NE procedure.

1. Introduction

An active line of research in the area of combinatorial optimization is concerned with developing various algorithms for a wide set of problems whose solutions are permutations. An important member of this set is the *single row facility layout problem* (SRFLP for short). Given a number of facilities and the flows between them, the SRFLP is to arrange the facilities along a straight line so as to minimize the total arrangement cost, which is the sum of the products of the flows and center-to-center distances between facilities. Suppose that there are n facilities having lengths $L_1, \ldots, L_n$, respectively. Let $W = (w_{ij})$ be a symmetric $n \times n$ matrix whose entry w_{ij} represents the flow of material between facilities i and j. Our intention in this paper is to deal with a version of the SRFLP where clearances between facilities, denoted as γ_{ij}, $i, j \in \{1, \ldots, n\}$, $i \neq j$, are not all equal to a (nonnegative) constant value. We emphasize that our approach to the SRFLP is applicable even when the matrix of clearances, $\Gamma = (\gamma_{ij})$, is not assumed to be necessarily symmetric. Because of this, and also to avoid losing the generality of the SRFLP formulation, we do not require symmetry in the matrix Γ. Certainly, the main diagonal of Γ is zero, and all other entries are nonnegative. With these notations, the SRFLP can be expressed as

$$\min_{p\in\Pi} F(p) = \sum_{k=1}^{n-1} \sum_{l=k+1}^{n} w_{p(k)p(l)} d_{p(k)p(l)}, \tag{1}$$

where Π is the set of all permutations of $\{1, \ldots, n\}$, $p(k)$ is the facility in the k-th position of permutation p, and $d_{p(k)p(l)}$ is the distance between the centroids of facilities $p(k)$ and $p(l)$. Let us assume that $k < l$. Then the distance is calculated according to the following equation:

$$d_{p(k)p(l)} = L_{p(k)}/2 + \sum_{\substack{m=k+1 \\ \text{if } l>k+1}}^{l-1} L_{p(m)} + L_{p(l)}/2 + \sum_{m=k}^{l-1} \gamma_{p(m)p(m+1)}. \tag{2}$$

We note that the formulation (1)–(2) was used by Datta, Amaral, and Figueira (2011) in their paper on a genetic algorithm approach to single row facility layout.

The SRFLP is a challenging research problem which has several real-life applications. In the area of flexible manufacturing systems, it models the linear layout of machines within manufacturing cells. In this type of layout, the machines are placed along a straight path travelled by an automated guided vehicle (Heragu & Kusiak, 1988). Other applications of the SRFLP include arranging a number of rooms on one side of a corridor in supermarkets, hospitals and office buildings (Simmons, 1969), arranging books on a shelf in a library (Picard & Queyranne, 1981), and design of warehouse layouts (Picard & Queyranne, 1981).

Because of the practical importance of the SRFLP, considerable attention has been given to the development of algorithms for its solution. Existing exact methods for the SRFLP include branch-and-bound (Simmons, 1969), dynamic programming (Picard & Queyranne, 1981), mixed-integer linear programming (Amaral, 2006, 2008; Heragu & Kusiak, 1991), cutting plane (Amaral, 2009), branch-and-cut (Amaral & Letchford, 2013), and semidefinite programming approaches (Anjos & Vannelli, 2008; Hungerländer & Rendl, 2013). Branch-and-bound

* Tel.: +370 37 300353.
E-mail address: gintaras@soften.ktu.lt, gintaras.palubeckis@ktu.lt

http://dx.doi.org/10.1016/j.ejor.2015.05.055
0377-2217/

(Palubeckis, 2012) and semidefinite programming (Hungerländer, 2014) algorithms were also applied for solving a special case of the problem in which all facilities have the same length. A computational comparison of the state-of-the-art exact methods for the SRFLP is given in the paper by Hungerländer and Rendl (2013). They report that the largest SRFLP instance solved to prove optimality involves 42 facilities. For the purpose of finding good but not necessarily optimal solutions for larger instances of the problem, a number of heuristic algorithms have been developed.

The fastest methods to generate feasible solutions for large-scale SRFLP instances are construction heuristics. Among them, a greedy-like algorithm of Heragu and Kusiak (1988) and an iterative construction procedure of Djellab and Gourgand (2001) can be mentioned. However, it is widely acknowledged that construction heuristics are not able to produce solutions of high quality. They can be applied in situations where computation time is a critical factor.

Another group of heuristic algorithms construct a permutation of facilities from the results obtained by solving either a mixed-integer or a semidefinite program (SDP). In particular, an algorithm relying on a mixed-integer programming model was presented by Heragu and Kusiak (1991). Recently, Amaral and Letchford (2013) have proposed an approach which allows obtaining a suboptimal solution as a byproduct of the branch-and-cut method. The crux of their approach is the use of a multi-dimensional scaling technique. Anjos, Kennings, and Vannelli (2005) were the first who proposed an SDP-based heuristic to produce a single row facility layout. The same strategy to obtain solutions to the SRFLP was followed by Anjos and Yen (2009) and Hungerländer and Rendl (2013).

The other way to approach the problem is to use metaheuristic search methods. The application of metaheuristics for the SRFLP dates back at least to Romero and Sánchez-Flores (1990) and Heragu and Alfa (1992), who developed simulated annealing algorithms for the problem. de Alvarenga, Negreiros-Gomes, and Mestria (2000) proposed another simulated annealing implementation for the SRFLP. The same authors also presented a tabu search algorithm and tested both metaheuristics on a small set of instances of size $n \leq 30$. More recent variants of tabu search strategy were proposed by Samarghandi and Eshghi (2010) and Kothari and Ghosh (2013a). Solimanpur, Vrat, and Shankar (2005) developed an ant algorithm for the SRFLP. Teo and Ponnambalam (2008) investigated a hybrid approach, combining ant colony optimization and particle swarm optimization (PSO) techniques. A pure PSO algorithm for the problem was proposed by Samarghandi, Taabayan, and Jahantigh (2010). Recently, Kothari and Ghosh (2013b) presented an insertion-based Lin–Kernighan heuristic for producing good quality layouts. The heuristic was shown to be competitive with other high-performance algorithms. There are also several layout methods available which follow the genetic paradigm. These include genetic algorithms of Ficko, Brezocnik, and Balic (2004) and Datta et al. (2011) as well as hybrid genetic algorithms of Ozcelik (2012) and Kothari and Ghosh (2014a). A similar evolutionary technique called the imperialist competitive algorithm was presented by Lian, Zhang, Gao, and Shao (2011). Single row layout algorithms based on the scatter search metaheuristic were proposed by Kumar, Asokan, Kumanan, and Varma (2008) and Kothari and Ghosh (2014b). The second of them was reported to yield very good solutions for popular benchmark SRFLP instances. For recent surveys on the single row facility layout problem, the reader is referred to Anjos and Liers (2012), Hungerländer and Rendl (2013), Keller and Buscher (2015), and Kothari and Ghosh (2012).

From the literature, it can be seen that many algorithms for the SRFLP incorporate a local search procedure (Amaral & Letchford, 2013; Heragu & Alfa, 1992; Heragu & Kusiak, 1991; Kothari & Ghosh, 2013a, 2014a, 2014b; Kumar et al., 2008; Ozcelik, 2012; Samarghandi & Eshghi, 2010; Samarghandi et al., 2010; Solimanpur et al., 2005; Teo & Ponnambalam, 2008). Two main types of local searches have been used for this problem (Kothari & Ghosh, 2013b). The first of them is based on pairwise interchanges of facilities (Amaral & Letchford, 2013; Heragu & Alfa, 1992; Heragu & Kusiak, 1991; Kothari & Ghosh, 2013a; Samarghandi & Eshghi, 2010; Samarghandi et al., 2010; Solimanpur et al., 2005; Teo & Ponnambalam, 2008), whereas the second one proceeds by executing insertion moves, where a facility is moved from one position to another in the permutation (Kothari & Ghosh, 2013a, 2014a, 2014b; Kumar et al., 2008; Ozcelik, 2012). The performance of local search (LS) algorithms greatly depends on the neighborhood exploration (NE) procedures (Kothari & Ghosh, 2013a). A straightforward implementation of such a procedure for each type of LS has time complexity $O(n^4)$ (Kothari & Ghosh, 2013a). Indeed, for example, in the case of interchange-based LS, there are $n(n-1)/2$ pairs of facilities, and computation of the objective function value for a permutation obtained by interchanging two facilities takes $O(n^2)$ operations. Recently, Kothari and Ghosh (2013a) developed NE procedures (for both interchange and insertion neighborhood structures) whose time complexity is $O(n^3)$. The algorithms of Kothari and Ghosh (2013a, 2013b, 2014a, 2014b) use these procedures and show good performance compared to other methods in the literature.

Many studies on the SRFLP (Amaral and Letchford, 2013; Anjos et al., 2005; Kothari and Ghosh, 2013a, among others) assumed that the clearance between each pair of facilities is equal to a constant value. In such a situation, by adequately adjusting the length of the facilities, zero clearances can be achieved. In this paper, however, our intention is to consider a more general model in which clearances between adjacent facilities are not necessarily all equal. There are two main reasons for such a choice. First, as emphasized by Solimanpur et al. (2005), allowing different clearances is important in real life manufacturing. Solimanpur et al. (2005) listed several factors that affect the clearance spaces required between facilities. They mentioned that an analytic approach, e.g. queuing models or simulation study, can be used to obtain the required data. Second, the assumption of different clearances between facilities adds no principal difficulties to our approach to constructing fast local search algorithms for the SRFLP.

The primary motivation of this paper is to develop more efficient neighborhood exploration algorithms than those presented by Kothari and Ghosh (2013a). We propose three NE procedures, one for searching pairwise interchange neighborhoods and the other two for searching insertion neighborhoods. The time complexity of each procedure is $O(n^2)$. We present empirical results comparing the performance of our NE procedures against those of Kothari and Ghosh (2013a). We embed these procedures in the variable neighborhood search algorithm for solving the SRFLP. We report on computational experiments on SRFLP instances of size up to 300 facilities.

The outline of the rest of the paper is as follows. In Section 2, we rephrase the objective function of the problem and introduce some preliminary notations. In Sections 3 and 4, we propose interchange-based and, respectively, insertion-based local search algorithms for the SRFLP. Their experimental evaluation is presented in Section 5. In Section 6, we provide a case study focused on the application of the variable neighborhood search metaheuristic for the considered problem. Concluding remarks are given in Section 7. Proofs of some results appear in Appendix A.

2. Preliminaries

The SRFLP (1)–(2) can be restated using an alternative form of the objective function. To present this form, we fix a permutation $p \in \Pi$ and consider a family of cuts induced by subsets of facilities $V_m = \{p(k) \mid k = 1, \ldots, m\}$, $m \in \{1, 2, \ldots, n-1\}$. Let $m \in \{1, 2, \ldots, n-1\}$. We call the sum $c_m = \sum_{k=1}^{m} \sum_{l=m+1}^{n} w_{p(k)p(l)}$ the *cut value* and the vector $C = (c_1, \ldots, c_{n-1})$ indexed by cuts the *cut vector*. We define $\lambda_r = L_r/2$ to be the half-length of the facility r, $r \in \{1, \ldots, n\}$. With these definitions, the objective function in (1) can be rewritten as follows.