APPLICATION OF LOCAL SEARCH METHODS FOR SOLVING A QUADRATIC ASSIGNMENT PROBLEM: A CASE STUDY

Gürdal Ertek[a *], Burak Aksu[b], Ş. İlker Birbil[c], Murat Cihan İkikat[d], Can Yıldırmaz[e]

Sabancı University
Orhanlı, Tuzla 34956
Istanbul, Turkey

{ertekg[a], sibirbil[c]}@sabanciuniv.edu
{buraksu[b], ikikat[d], canyildirmaz[e]}@su.sabancuniv.edu

Abstract:
This paper discusses the design and application of local search methods to a real-life application at a steel cord manufacturing plant. The case study involves a layout problem that can be represented as a Quadratic Assignment Problem (QAP). Due to the nature of the manufacturing process, certain machinery need to be allocated in close proximity to each other. This issue is incorporated into the objective function through assigning high penalty costs to the unfavorable allocations. QAP belongs to one of the most difficult class of combinatorial optimization problems, and is not solvable to optimality as the number of facilities increases. We implement the well-known local search methods, 2-opt, 3-opt and tabu search. We compare the solution performances of the methods to the results obtained from the NEOS server, which provides free access to many optimization solvers on the internet.

Keywords:
quadratic assignment problem, steel cord manufacturing, local search methods, tabu search, NEOS online server

Introduction

This paper discusses local search heuristics applied to a real-world problem from industry. The problem is the determination of the layout of a steel-cord manufacturing factory. Steel cord is typically used as the main reinforcement material in manufacturing steel radial tires. It strengthens the tire to provide fuel savings, long mileage, safety and comfort. The steel cord manufacturing goes through continuous processes, where wire semi-products are stored on discrete inventory units, namely "spools" (see Figure 1). The literature on steel cord manufacturing is not extensive, since this is a very specialized type of manufacturing, and the systems required are produced and installed by only a handful of companies in the world. We refer the interested readers to the following three studies: Thomas *et al.* (2002) report improvement of operations in a steel cord manufacturing company using simulation. Mercankaya (2003) develops an optimization-based decision support system for steel cord manufacturing. Türkseven and Ertek (2003) explain how the quality and the productivity were improved in steel cord manufacturing through custom-built simulation software. Their objective is to determine the optimal spool lengths under certain constraints on the spool lengths.

---

[*] Corresponding author

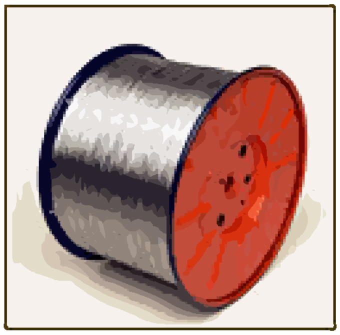


Figure 1. Spool on which wire is wound

In steel cord manufacturing incoming raw material, the “steel rod wire”, is thinned by dry and wet drawing into “filaments” that are used in successive bunching operations to construct the “steel cord” final products (see Figure 2). The focus of our research is the second phase of production, which starts with wet wire drawing and ends with spiraling. The first phase of the production is carried out by machines that are fixed to their locations. The chances of moving these machines are next to none, since considerable time and resources would be required to make such movements. The second phase of production, on the other hand, is carried out by machines that can be relocated. The only important issue with the second phase of production is that certain machinery, which we refer to as machine types MT01, MT02 and MT03, have to be located in the neighborhood of a lubricant pool. These machines carry out wet drawing of steel cord, and use the lubricant liquid, which is supplied to the machines by an underground pipeline system. We reflected this location constraint in the mathematical model that we developed by assigning a high flow volume between these machine types and the lubricant cells.

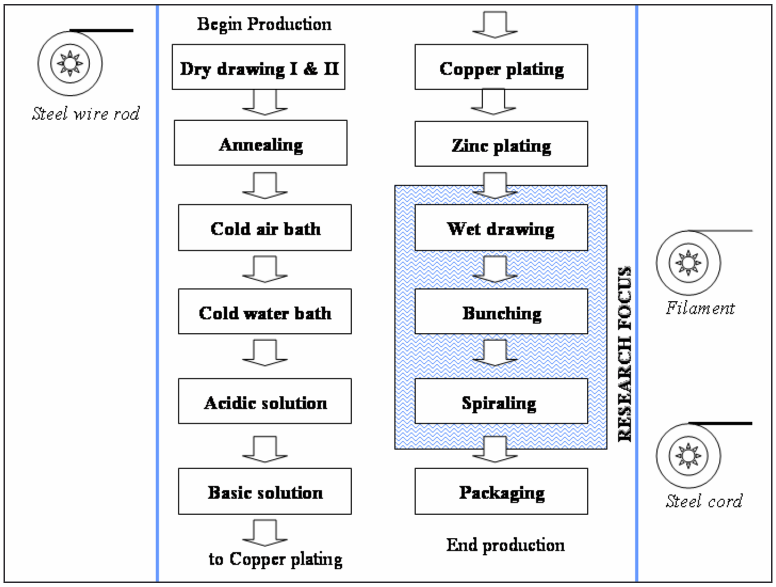


Figure 2. Production processes in steel cord manufacturing

The Mathematical Model

We assume that the flow from an area of machine type i to another area of machine type j is equally distributed (see Figure 3). Notice that this is a simplification of the actual process because after the machines are assigned to the locations, one would send as much flow as possible from a type i machine to the closest type j machine. We elaborate this issue in the summary and future research section.

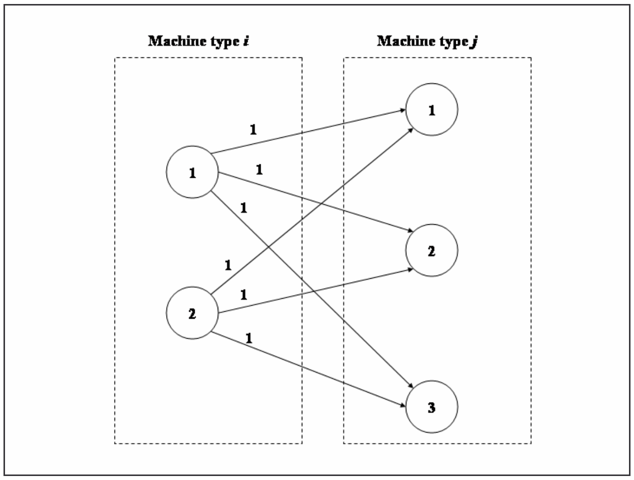


Figure 3. Equal distribution of flow from an area of machine type i to another area of machine type j

In the subsequent discussion, the acronyms LBR and WH stand for the lubricant pool and the warehouse, respectively. Before we give the mathematical model, let us define the sets, the parameters and the decision variables.

Sets

M: the set of machine types, M = {MT01, MT02, …, MT12, LBR, WH}

A: the set of areas, A = {1, 2, …, 69}

Parameters

F_{ij}: the total flow from an area dedicated to machine type i to an area dedicated to machine type j; $i, j \in M$

D_{kl}: the rectilinear distance between area k and l; $k, l \in A$

K_i: the number of areas assigned to machine type i

Decision variables

$x_{ik} = 1$ if area k is assigned to machine type i; 0 otherwise.

The overall mathematical model then becomes

$$\min \quad \sum_{i\in M}\sum_{k\in A}\sum_{j\in M}\sum_{l\in A} F_{ij} D_{kl} x_{ik} x_{jl}$$

$$\text{s.t.} \quad \sum_{k\in A} x_{ik} = K_i, \qquad \forall i \in M \qquad (1)$$

$$\sum_{i\in M} x_{ik} = 1, \qquad \forall k \in A \qquad (2)$$

$$x_{ik} = 1, \qquad \forall (i,k) \in \{(i,k) : i = \text{LBR}, k = 1,2,\cdots,6\} \qquad (3)$$

$$x_{ik} = 1, \qquad \forall (i,k) \in \{(i,k) : i = \text{WH}, k = 49,54,59,64,69\} \qquad (4)$$

$$x_{ik} \text{ binary} \qquad \forall i \in M \text{ and } \forall k \in A. \qquad (5)$$

The constraint set (1) ensures that the number of areas allocated to each machine type is equal to K_i; the required number of areas for that machine type. The constraint set (2) ensures that each area is assigned to exactly one machine type. To fix the areas assigned to the lubricant and to the warehouse, we introduce the constraint sets (3) and (4), respectively. The last set of constraints, (5) states that the decision variables x_{ik} should be binary.

As we mentioned above, to locate the machine types MT01, MT02 and MT03 close to the lubricant pool, we have assigned a large flow value between these machine types and the lubricant pool. We also assigned high values for flows from machines that directly feed into the warehouse (namely, machines MT04, MT06, MT07, MT11, MT12). Clearly, assigning unnecessarily large values for specific flows creates sudden jumps in the objective function values. Therefore, a special care should be taken to set this value. In our study, we have used the following calculation

$$F_{ij} = F_{ji} = \overline{F} \text{ for } i = \text{LBR} \text{ and } j \in \{\text{MT01}, \text{MT02}, \text{MT03}\},$$

$$F_{ij} = F_{ji} = \overline{F} \text{ for } i = \text{WH} \text{ and } j \in \{\text{MT04}, \text{MT06}, \text{MT07}, \text{MT11}, \text{MT12}\},$$

where

$$\overline{F} = \sum_{i \in M \setminus \{LBR, WH\}} \sum_{j \in M \setminus \{LBR, WH\}} F_{ij}.$$

The presented mathematical model is very similar to the Quadratic Assignment Problem (QAP) problem. Thus, some of the solution methods developed for QAP can be applied to solve the presented problem after minor modifications. Among these solution methods, local search heuristics such; as swapping, tabu search, and so on, are frequently used because these methods are relatively easy to implement. Moreover, due to the moderate size of the presented problem, standard solvers may also provide a solution. This is particularly important for practitioners, who prefer to use software packages rather than creating their own tools. We next present our efforts to solve the problem with an easily accessible online solver as well as with some local search methods.

Solution Approaches

GAMS Model

The first solution approach that we applied was to build a mathematical model using the GAMS software (http://www.gams.com). Since the model required specification of the rectilinear distances between areas we had to implement a "model generator" program to generate the model automatically based on input data. This implementation was carried out using the Java programming language. The model generator reads in the machine types, the number of areas assigned to each machine type (K_i), and the flows from each area of machine type i to each area of machine type j ($i, j \in M$). The model generator then generates the GAMS model file which is submitted to NEOS server which provides free access to many optimization solvers on the internet (http://www-neos.mcs.anl.gov/neos/).

We used the submit client (written in Java) to submit the GAMS model to NEOS server. While submitting we selected the Solvers menu, the menu item "Mixed Integer Nonlinearly Constrained Optimization", and finally the menu item "SBB [GAMS Input]".

The solution found by the online solver has objective function value of 6,953,483 and is given in Figure 4. The visualization is done by a Java program. In this solution MT01, MT02, and MT03 are assigned to areas close to the lubricant (LBR).

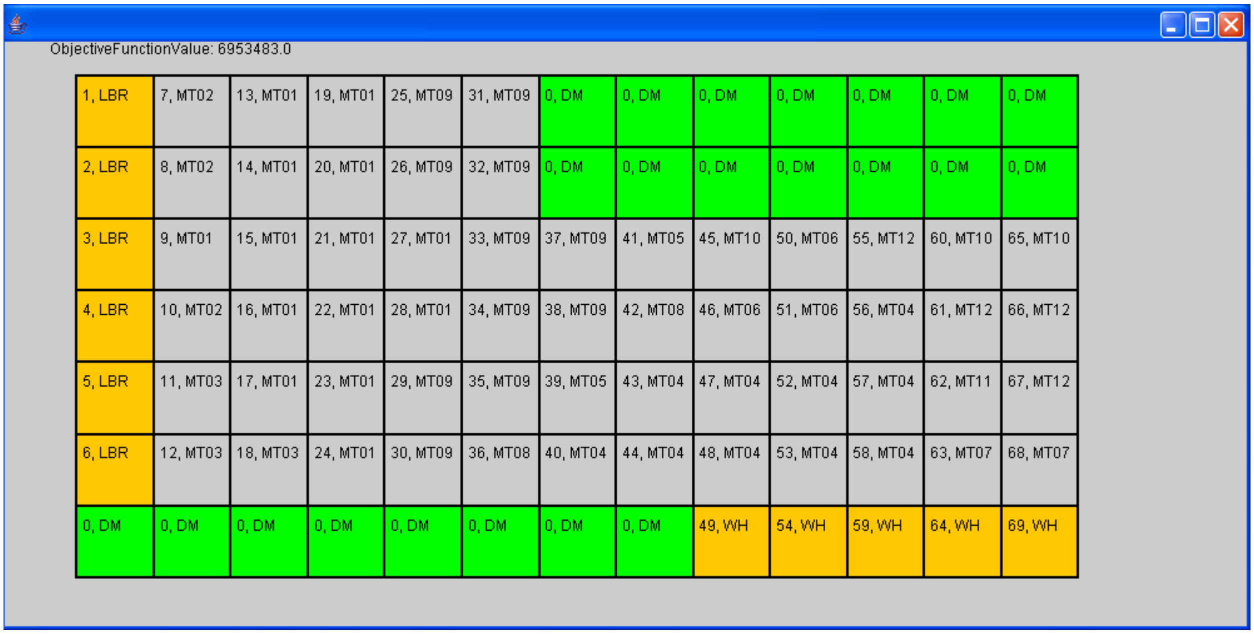
2-opt and 3-opt

We implemented 2-opt and 3-opt algorithms for the problem again using Java. The 2-opt algorithm performs an exchange among the machine types of two areas, and keeps the solution if the objective function (OF) value has improved (decreased). 3-opt performs an exchange among machine types of three areas, and keeps the solution if the OF value has improved (decreased).

We fed the solution in Figure 4 as the starting solution for both 2-opt and 3-opt. The algorithms terminated without finding any improved solutions.

Tabu search

Finally we employed the well-known tabu search heuristic developed by Taillard (1991) to our problem. The C++ code for the heuristic is available at the following address: http://ina.eivd.ch/collaborateurs/etd/. Since the objective function value in our problem has to involve the additional costs resulting from the large flows to the lubricant cells, we have slightly modified the algorithm proposed by Taillard (1991). We selected 5 different starting solutions, and observed that all these solutions had the objective function value of 6,953,483, which is identical to the objective function value found by NEOS server. This shows that tabu search heuristic is very much applicable for the problem that we have presented.



ObjectiveFunctionValue: 6953483.0

1, LBR	7, MT02	13, MT01	19, MT01	25, MT09	31, MT09	0, DM	0, DM	0, DM	0, DM	0, DM	0, DM	0, DM
2, LBR	8, MT02	14, MT01	20, MT01	26, MT09	32, MT09	0, DM	0, DM	0, DM	0, DM	0, DM	0, DM	0, DM
3, LBR	9, MT01	15, MT01	21, MT01	27, MT01	33, MT09	37, MT09	41, MT05	45, MT10	50, MT06	55, MT12	60, MT10	65, MT10
4, LBR	10, MT02	16, MT01	22, MT01	28, MT01	34, MT09	38, MT09	42, MT08	46, MT06	51, MT06	56, MT04	61, MT12	66, MT12
5, LBR	11, MT03	17, MT01	23, MT01	29, MT09	35, MT09	39, MT05	43, MT04	47, MT04	52, MT04	57, MT04	62, MT11	67, MT12
6, LBR	12, MT03	18, MT03	24, MT01	30, MT09	36, MT08	40, MT04	44, MT04	48, MT04	53, MT04	58, MT04	63, MT07	68, MT07
0, DM	0, DM	0, DM	0, DM	0, DM	0, DM	0, DM	0, DM	49, WH	54, WH	59, WH	64, WH	69, WH

Figure 4. The layout suggested by both NEOS and local search heuristics

Summary and Future Research

We have presented a slightly-modified QAP model and discussed the solution approaches that we employed to solve the model. For our problem, where the real-world data came from a steel cord manufacturer, the solution (in Figure 4) suggested by a well-known tabu search heuristic was the same as the solution found by the NEOS server. We have also observed that 2-opt and 3-opt heuristics could not improve on this solution. This solution thus can be considered a pseudo-optimal solution.

There is an interesting research problem that can be defined based on the problem described here. In the problem that we presented we have assumed that the flow from each area of machine type *i* to each area of machine type *i* is equal. There is no guarantee that this is the best policy. Indeed, the decision of *how to assign the flows* from each area of machine type *i* to each area of machine type *j* is an embedded network flow problem. For example, the optimal flows for the system in Figure 3 can be as given in Figure 5. Given a layout, each flow assignment is a possible solution, and given a flow assignment each layout choice is a possible solution. Thus, there is a need to develop algorithms that can solve these two interrelated problems simultaneously. Birbil et al. (2005) discuss this problem.

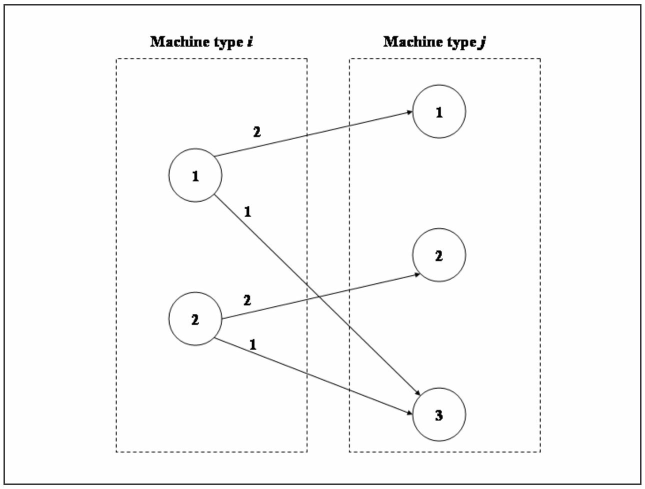


Figure 5. A possible assignment of flows between areas of two machine types

References

Birbil, Ş. İ., Ertek, G., Frenk, J. B. G., Still, G., & Zhang, S. (2005). Simultaneous solution of assignment and network flow problems. *Working paper*. Istanbul, Turkey: Sabancı University.

Mercankaya, B. (2003). Sales forecasting and production planning of BEKSA in the context of supply chain management. *LFI report*. Istanbul, Turkey: Sabancı University.

Taillard, E. (1991). Robust taboo search for the QAP. *Parallel Computing*, 17, 443-455.

Thomas, J., Todi, J., & Paranjpe, A. (2002). Optimization of operations in a steel wire manufacturing company. In E. Yücesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes: Proceedings of 2002 Winter Simulation Conference (pp. 1151-1156). Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Türkseven, C. H., & Ertek, G. (2003). Simulation modeling for quality and productivity in steel-cord manufacturing. In S. Chick, P. J. Sanchez, D. Ferrin, & D. J. Morrice: Proceedings of 2003 Winter Simulation Conference (pp. 1225-1229). Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.