

Two optimal algorithms for finding bi-directional shortest path design problem in a block layout

*M. Hamzeei**

M.Sc., Dep. of Industrial Engineering, Sharif University of Technology, Tehran, Iran

R. Zanjirani Farahani

Assistant Professor, Dep. of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran

Abstract

In this paper, Shortest Path Design Problem (SPDP) in which the path is incident to all cells is considered. The bi-directional path is one of the known types of configuration of networks for Automated Guided Vehicles (AGV). To solve this problem, two algorithms are developed. For each algorithm an Integer Linear Programming (ILP) is determined. The objective functions of both algorithms are to find the shortest path. The path must be connected and incident to all cells at least in one edge or node. A simple Branch-and-Cut approach is used to solve the ILP models. Computational results show that the models easily can solve the problem with less than 45 cells using a commercial ILP solver.

Keywords: AGV; Block layout; Bi-directional path; Integer Linear Programming, Branch-and-Cut

1. Introduction

Facility Planning (FP) as one of the most important task of industrial engineering branches, is one of the eldest activities of industrial engineers. Facility location and facility layout are two sections of FP. Facility layout involves in determining the department layout, production and manufacturing units, stores and so on. Tompkins et al. [29] also stated that facility layout includes layout design, material handling system design and facility system design. Apple [2] pointed out that it is crucial to incorporate material handling decisions in layout design. Tompkins et al. [29] estimated that 20 to 50 percent of production costs are related to layout design and material handling. Therefore any cost reduction in this area will lead to cost reduction in a production unit.

In a layout design, each cell is represented by a rectilinear. This cell is not necessarily a convex polygon. A block layout is a set of fully packed of these cells. Any planning and analysis in a block layout is impossible without considering material handling system. American Society of Material Handling defined material handling as the art and knowledge which involves in movement, packing and storing of materials in every shapes and forms. Tanchoco and Sinriech

[28] defined material flow system in terms of material handling equipments, configuration and direction of material handling networks, and number and location of pick-up and delivery stations. Among all types of material handling equipments, in this paper the authors consider Automated Guided Vehicle (AGV).

AGVs are driverless vehicles used for transporting materials and goods within a plant layout. They usually follow a wire guided-path. (see, e.g., Hodgson, King and Monteith [14]). When there is more than one vehicle, a controller is responsible for regulation of traffic. In recent years, the application of AGVs was increased as a horizontal material handling equipment in production plants.

One of the common problems in AGV systems is finding a route that serves all cells in a block layout. This problem was first studied by Maxwell and Muckstadt [19]. They introduced a model to determine maximum number of vehicles in a plant layout to efficiently transfer material from one department to another. In their model, they assumed that the best routes had been found and installed.

There are some types of material handling network configurations which have been considered by the researchers. These types are (1) Conventional configuration, (2) Unidirectional single loop, (3) Tandem

* Corresponding author. E-mail: m_hamzehie@ie.sharif.edu

configuration, (4) Segmented Flow Topology (SFT) and (5) Bi-directional path. Conventional configuration was modeled by Gaskin and Tanchoco [13], Kaspi and Tanchoco [15] and Seo and Egbelu [22] among the others who used mathematical programming. In conventional configuration, all edges are parts of the material handling network and the models are to determine the direction of the edges. Unidirectional single loop problem determines a loop which serves all pick-up and delivery stations. This type of material handling networks has been studied by several researchers like Tanchoco and Sinriech [28], Sinriech and Tanchoco [23,24], Laporte et al. [17], Aef-Vaziri et al. [3,4] and Farahani et al. [11]. Tandem configuration involves breaking up the entire block layout into non-overlapping and separated cells and assigning a single loop to each cell. Each cell is served by a single unidirectional AGV. Separated cells are connected to each other with a set of conveyors [6]. Bozer and Srinivasan [5,7] used a partitioning algorithm based heuristic to divide the shop floor into various tandem loops. SFT divides block layout into several parts. An AGV is assigned to each part. Sinriech and Tanchoco had some studies on SFT at 1994, 1995 and 1997 [25,26,27]. Another type of material handling networks is bi-directional path. Figure 1 shows a path in a basic block layout.

Egbelu and Tanchoco [10] studied the bi-directional path in 1986. Although in a bi-directional path, traffic flow takes place in either direction in each aisle, however vehicles are not allowed to travel in opposite directions at the same time. Therefore, buffer areas should exist for temporary parking of vehicles [10]. Egbelu and Tanchoco [10] discussed about types of buffers, place and number of required buffers. They also introduced a model which describes the flow and control of AGVs in a bi-directional network. With simulation, it has been shown that in a specified situation, the use of bi-directional guide-paths in networks with few AGVs can lead to an increase in productivity. Kim and Tanchoco [16] also presented simulation results to compare the performance of unidirectional and bi-directional layouts in a particular network. For this network, it has been shown that the bi-directional layout outperforms the unidirectional one in terms of the number of jobs completed per time unit. Chhajed et al. [8] had another study on bi-directional path in 1992. In 2004, Rajagopalan et al. [20] discussed two models considering material flow system in addition to bi-directional path. Their models addressed material flow path and the location of P/D stations, but they did not ensure optimal solution. Also in these models, both loaded and unloaded vehicles were con-

sidered. In the first model flow path for both loaded and unloaded vehicles was the same, but in the second model, the flow path for loaded and unloaded could be different.

In this article we consider the Shortest Path Design Problem (SPDP). The assumptions are (1) cell layout is known (2) flow path is bi-directional (3) P/D stations are not considered. P/D stations could be on the edge or nodes of cells which are adjacent to path. We use two Integer Linear Programming (ILP) models to solve this problem. Input of these models is only block layout and the objective functions are to minimize total path length. De Gazman [9] has proved that this problem is NP-hard.

Afentakis [1] stated the advantages of loop pattern for material handling network configuration as simplicity and efficiency, low initial and expansion cost and flexibility in process and production. For bi-directional path in addition to loop advantages, there are some advantages which managers prefer to use. A path usually has a shorter length than a loop, so it used less space of a production plant for aisles. Also shorter length will result in less time to travel. Sometimes when a single loop is impossible to form in a special block layout, a bi-directional path is very easy to implement.

In Section 2 we will have a definition of the problem. Section 3 involves in mathematical model for the algorithm. For each algorithm, an ILP is developed and then an algorithm is introduced based on Branch-and-Cut approach to solve the problem. Computational results will be presented in Section 4 and eventually Section 5 contains conclusion and future research suggestions.

2. Problem definition

The problem of designing shortest path in a block layout is considered. Suppose that the production plant is divided into n cells. Each cell is a rectangle, not necessarily a rectangular but a rectilinear with angles of 90, 180 and 270 degrees. So each cell may be not a convex polygon.

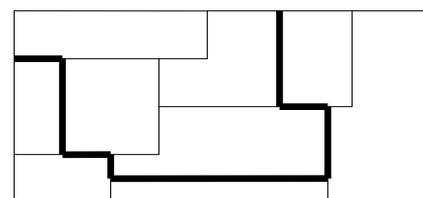


Figure 1. Bi-directional path in a block layout.

As pointed out previously, these models have block layout as an input. To explain this problem, we consider a prototype example used by Farahani et al. [11]. The block layout is shown in Figure 2.

There are two versions for path in SPDP (i) the path must be incident with each cell at least in one side (Figure 3) and (ii) the path must be adjacent to each cell at least in one node (Figure 4). The proposed algorithm is able to solve both versions.

A feasible path is one that is connected and is adjacent to each cell at least in one edge or one node. If one of these conditions is violated, then the path will be infeasible. Figures 5 and 6 show cases that each condition violates.

It may be a block layout which has not any feasible path. Figure 7 shows a block layout without any feasible path.

A block layout could be mentioned with an adjacency graph in which each node represents a cell and each edge represents adjacency of the cells. Figure 8 shows the adjacency graph and adjacency matrix of the block layout of Figure 2.

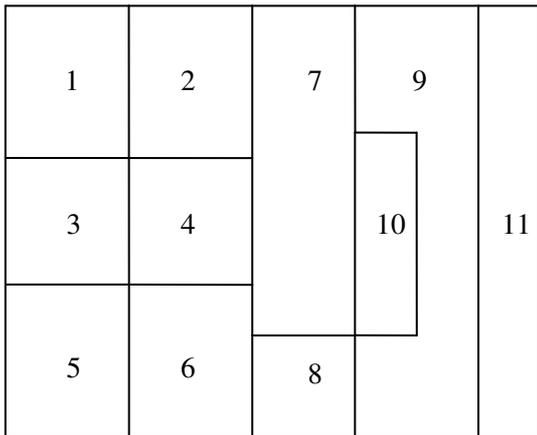


Figure 2. A block layout (Farahani et al. [11]).

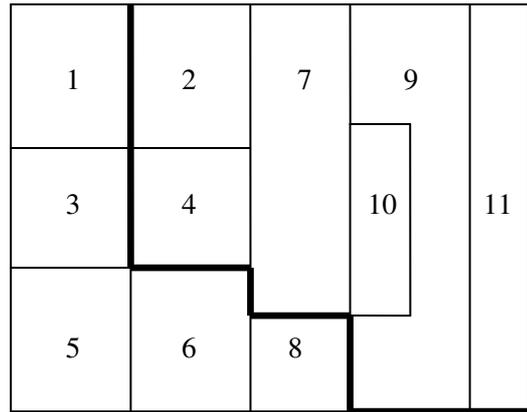


Figure 4. A path which is adjacent to each cell at least in one node (cell #10 is adjacent to the path in only one node).

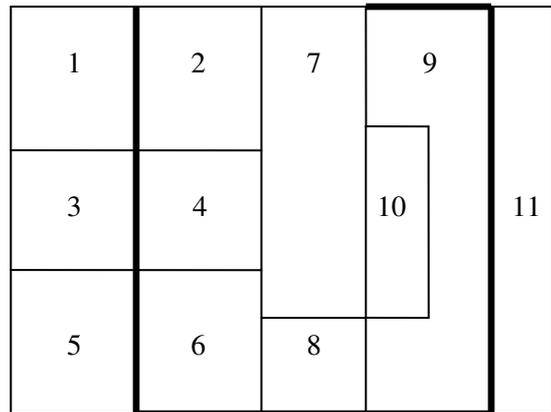


Figure 5. Some cells are not adjacent to the path.

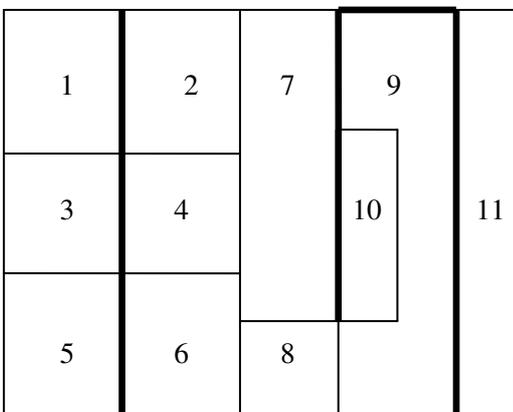


Figure 3. A path which is adjacent to each cell at least in one edge.

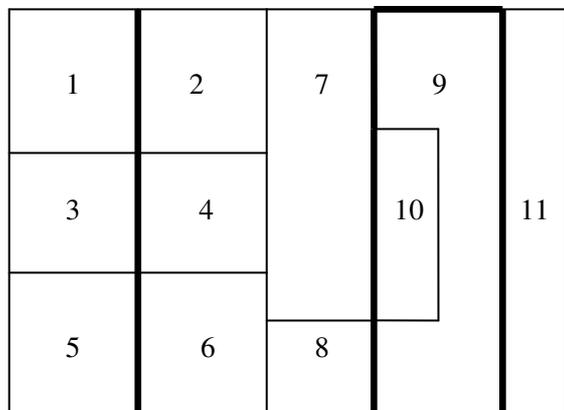


Figure 6. The path is not connected.

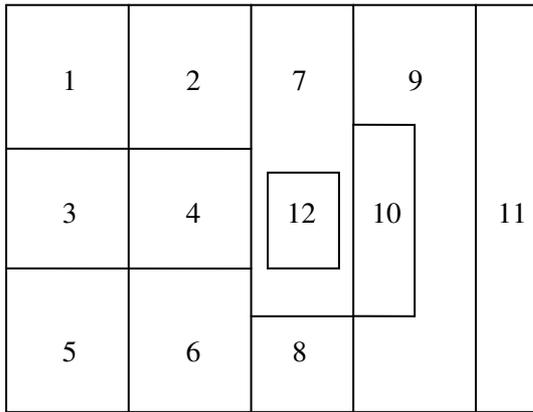
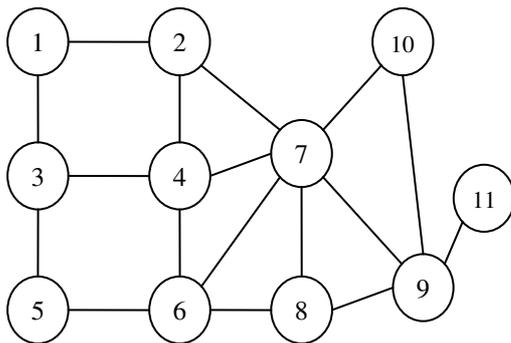


Figure 7. A block layout without any feasible path.



(a)

0	1	1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0
0	1	1	0	0	1	1	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0
0	0	0	1	1	0	1	1	0	0	0
0	1	0	1	0	1	0	1	1	1	0
0	0	0	0	0	1	1	0	1	0	0
0	0	0	0	0	0	0	1	0	1	1
0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	0	1	0	0

(b)

Figure 8. (a) Adjacency graph, (b) Adjacency matrix.

SPDP is shown with unidirectional graph $G = (V, E)$. V is the set of nodes and E is the set of edges. In our notation, if (i, j) is an edge, then i is less than j ($i < j$). Asef-Vaziri et al. [3] considered Shortest Loop Design Problem (SLDP) and developed a model to solve this problem. They stated that degree 2 nodes do not have any role in solving the problem. But here, in our problem SPDP, degree 2 could have an effective role. Figure 9 shows a feasible path, but in this path a degree 2 node exists. Therefore we consider entire node, degrees 2, 3 and 4, during running the algorithms.

SLDP is related to Generalized Traveling Salesman Problem (GTSP) [3]. GTSP is involved in finding shortest Hamiltonian cycle in certain sets of nodes in which the cycle must be adjacent one node of each set. When each set includes only one node, GTSP converts to Traveling Salesman Problem (TSP). GTSP is formulated and solved using ILP by Laporte and Nobert [18] and Fischetti et al. [12]. Asef-Vaziri et al. stated two differences between GTSP and SLDP. First, the constraints of SDLP are to cover edges, but GTSP is to cover nodes and second, connectivity constraints in SLDP are much more less than GTSP.

SPDP is very similar to SLDP. When we use a virtual node in block layout graph, every loop that includes the virtual node, actually is a path. Therefore SPDP and SLDP can easily convert to each other and SPDP also is similar to GTSP.

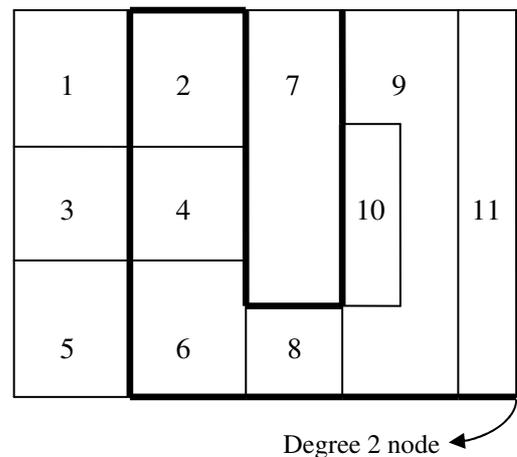


Figure 9. A feasible path with a degree 2 node.

3. SPDP

We use Integer Linear Programming to solve the SPDP. In our algorithms, we use two different ILPs. The first ILP use a very clear specification of the path. A path is a tree. In this tree, all nodes are degree 2 except 2 nodes which are degree 1. One of the attributes of a tree is that the number of nodes is less than the number of edges by one. This is the basic concept of the first formulation. The second formulation uses the formulation which has been used with Aset-Vaziri *et al* [3]. They have solved SLPD, but we add a constraint to their model to insert a virtual node permanently in the loop. This loop is actually a path. Other completing explanations are as follow.

3.1. First algorithm

As it was explained previously, in this algorithm an ILP used that it is based on an attribute of trees.

3.1.1. Sets and indices

- $C = \{1, 2, 3, \dots, n\}$: Set of cells in a block layout graph ($p \in C$).
- $S \subset C$: Subset of C .
- $\bar{S} = C \setminus S$: Full complement of S .
- $V = \{1, 2, \dots, v\}$: Set of nodes in block layout graph ($i, j, k, l \in V$).
- $V_p = \{1, 2, \dots, v_p\}$: Set of nodes of cell p in block layout graph ($\forall p \in C, \bigcup_{p \in C} V_p = V$).
- $E = \{1, 2, 3, \dots, e\}$: Set of nodes in block layout graph ($(k, l) \in E \mid k, l \in V$ are adjacent nodes).
- $E_p = \{1, 2, 3, \dots, e_p\}$: Set of edges of cell p in block layout graph ($\forall p \in C, \bigcup_{p \in C} E_p = E$).
- $E(S) = \{(i, j) \in E \mid a \in S : (i, j) \in a\}$: Set of all edges belong to subset S in block layout graph.

Now, a relation is defined as follows:

$$A_{ab} = E_a \cap E_b, \forall a, b \in C. \quad (1)$$

This relation defines a set which includes common edges between two subsets of S . According to relation (1), the following subsets are defined:

$S_A = \{S \subset C \mid \forall a \in S : \sum_{b \in S} |A_{ab}| \geq 1\}$: Set of all subsets of adjacent cells in block layout graph.

S_{A_m} : One member of S_A .

$B(S_A) = \{(i, j) \in E(S_A) \mid \exists b \in \bar{S}_A : (i, j) \in E_b\}$: Set of edges on boundary of S_A .

3.1.2. Variables and parameters

c_{ij} Length of edge (i, j) . ($(i, j) \in E \mid i, j \in V$ are adjacent nodes).

x_{ij} Binary variable which is equal to 1 if and only if (i, j) is adjacent to path and 0 otherwise. ($(i, j) \in E \mid i, j \in V$ are adjacent nodes).

y_k Binary variable which is equal to 1 if and only if node k is adjacent to path and 0 otherwise.

v_k Binary variable which is equal to 1 if and only if node k is at the start or finish node of path and 0 otherwise.

3.1.3. Mathematical model

SPDP could be solved using a model as follow:

$$\text{Min} \sum_{i < j} c_{ij} \cdot x_{ij} \quad (2)$$

Subject to:

$$\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2y_k - v_k \quad (k \in V) \quad (3)$$

$$\sum_{(i, j) \in E_p} x_{ij} \geq 1 \quad (p \in C) \quad (4)$$

$$\sum_{(i, j) \in B(S_{A_m})} x_{ij} \leq |B(S_{A_m})| - 1 \quad (S_{A_m} \in S_A) \quad (5)$$

$$v_k \leq y_k \quad (k \in V) \quad (6)$$

$$\sum_{(i,j) \in E} x_{ij} = \sum_{k=1}^v y_k - 1 \quad (7)$$

$$y_k = 0 \text{ or } 1 \quad (k \in V) \quad (8)$$

$$x_{ij} = 0 \text{ or } 1 \quad ((i, j) \in E) \quad (9)$$

$$y_k = 0 \text{ or } 1 \quad (k \in V) \quad (10)$$

Relation (2) is the objective function and is to minimize total length of the path.

Relation (3) is the degree constraint. This relation is very similar to relation which was used by Asef-Vaziri et al. [3] for degree constraint. This means if a node is in the middle of path, two adjacent edges of it must be in the path. If a node is the start or finish node of the path, then only one of its adjacent edges could be in the path. If a node is not in the path, then none of its adjacent edges could be in the path.

Relation (4) is covering constraint. This relation is similar to covering constraint in [3]. But if SPDP assumes a path feasible when it is adjacent in at least one node, we can use another constraint like this:

$$\sum_{k \in V_p} y_k \geq 1 \quad p \in N. \quad (11)$$

Relation (5) is tour eliminator constraint. This relation has a similar role to connectivity constraint in [3].

Relation (6) states that a node could be in start or finish node of path if it is in the path. Relation (7) keeps the selected edges being a tree. Relations (8), (9) and (10) are integrality constraints.

Lemma. In this formulation, relation (6) will be relaxed.

Proof. The right side of constraint (3) is summation of non-negative variable. So it is always greater than or equal to zero. Therefore the left side will be greater than 0 or equal to. If the left side is equal to 0 then:

$$2y_k - v_k = 0 \Rightarrow 2y_k = v_k \Rightarrow y_k = v_k = 0.$$

Otherwise if the left side is greater than 0 then:

$$2y_k - v_k > 0 \Rightarrow 2y_k > v_k.$$

Now if $v_k = 0$ then ($y_k = 0$ or 1) and if $v_k = 1$ then $y_k = 1$.

It is obvious that in all states, v_k is equal or less than y_k for all values of k . Therefore the proof is complete.

3.2 Second algorithm

In the second algorithm like first algorithm for each problem an ILP should be solved. In this ILP we used the concepts which are used with Asef-Vaziri et al. [3] for solving SLDP. When a virtual node is always in the loop, this loop will be a path. Figure 10 shows a block layout and a virtual node. As you can see, a loop in presence of virtual node is visible. But it is actually a path. We consider a virtual cell in block layout. This virtual cell has only one node. In our notation we use index 0 for virtual cell and node.

3.2.1. Sets and indices

The sets and indices for this algorithm are very similar to those of first algorithm, but there are some differences which should be explained. Therefore we define them again.

- $C = \{1, 2, 3, \dots, n\}$: Set of all cells in block layout graph.
- $N = C \cup \{0\} = \{0, 1, 2, \dots, n\}$: set of all cells and virtual cell in block layout graph and virtual cell ($p \in N$).
- $S \subset C$: A subset of C .
- $\bar{S} = N \setminus S$: The full complement set of S .
- $V = \{0, 1, 2, \dots, v\}$: Set of all nodes and virtual node in block layout graph ($i, j, k, l \in V$).
- $V_p = \{1, 2, \dots, n_p\}$: Set of nodes of cell p ($\forall p \in N, \cup_{p \in N} V_p = V$).
- $E = \{1, 2, 3, \dots, e\}$: Set of all edges in block layout graph and virtual edges ($((k, l) \in E \mid k, l \in V$ are adjacent nodes).
- $E_p = \{1, 2, 3, \dots, e_p\}$: Set of all edges of cell p ($\forall p \in N, \cup_{p \in N} E_p = E$).
- $E(S) = \{(i, j) \in E \mid a \in S : (i, j) \in E_a\}$: Set of all edges of set S in block layout graph.

A_{ab} , S_A , S_{A_m} and $B(S_A)$ are defined like section 3.1.1.1.

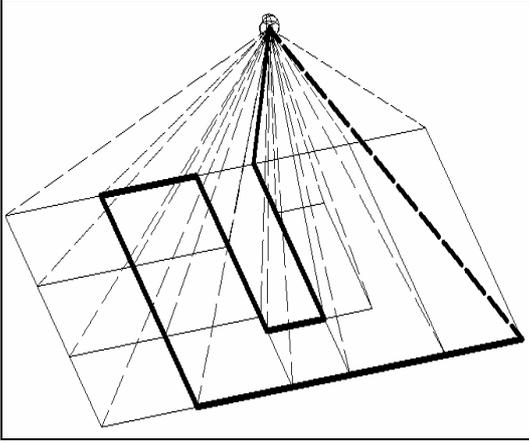


Figure 10. A block layout and virtual node.

3.2.2. Variables and parameters

For this model, variables and parameters are like those of the first algorithm. c_{kl} , x_{ij} , y_k and v_k exactly are the same with those which are defined in section 3.1.2.

3.2.3. Mathematical model

The following model could be used for solving SPDP:

$$\text{Min } \sum_{i < j} c_{ij} \cdot x_{ij} \quad (12)$$

Subject to:

$$\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2y_k \quad (k \in V) \quad (13)$$

$$\sum_{(i,j) \in E_p} x_{ij} \geq 1 \quad (p \in N, p \neq 0) \quad (14)$$

$$\sum_{(i,j) \in B(S_{A_m})} x_{ij} \leq |B(S_{A_m})| - 1 \quad (S_{A_m} \in S_A) \quad (15)$$

$$y_k = 0 \text{ or } 1 \quad (k \in V) \quad (16)$$

$$x_{ij} = 0 \text{ or } 1 \quad ((i,j) \in E) \quad (17)$$

$$y_0 = 1 \quad (18)$$

Relation (12) is like relation (2) and the objective function. Relation (13) is like relation (3). It is possi-

ble to change this relation with the following constraint [3]. Therefore variable y_k can be removed from the formulation.

Relation (14) is like relation (4) covering constraint. This relation is similar to covering constraint in [3]. But if SPDP assumes a path feasible when it is adjacent in at least one node with each cell, then we can use another constraint like relation (11) for this purpose. Relation (15) is like relation (5). Relation (16) and (17) is integrality constraint. Relation (18) states that virtual node has to be in the loop.

4. Algorithm

As it was mentioned before, our attempt is to prepare the algorithm for solving the problem with either assumptions at least one edge or one node. When a feasible path has to be adjacent with at least one edge, then constraints (3), (4), (5), (7), (8), (9) and (10) would be enough. But otherwise constraints (3), (5), (7), (8), (9), (10) and (11) would be used.

Relation (5) is the root cause of difficulty in solving this problem because it is exponential in n . Therefore we use a simple Branch-and-Cut approach to solve this problem in our algorithm. This makes large number of constraints relax initially and introduce in a dynamic fashion as violations are detected. This simplified version of this approach which does not need any programming and could work on LINGO [21] is showed below (relation numbers which are in brackets relate to second algorithm):

Step 1. If the path for feasibility needs to be incident to each cell only at least in one node go to Step 5.

Step 2. Set $c := 1$ as iteration count. Initialize a linear problem in which constraints (3), (4) and (7) [(13) and (14)] are introduced.

Step 3. Solve the model with LINGO 8.00. If the solution is feasible, stop.

Step 4. Find members of set S_A for violated constraints (5) [(15)]. Introduce the constraints (5) [(15)] for these members. Set $c := c + 1$. Go to Step 3.

Step 5. Set $c := 1$ as iteration count. Initialize a linear problem in which constraints (3), (7) and (11) [(11) and (13)] are introduced.

Step 6. Solve the model with LINGO 8.00. If the solution is feasible, stop.

Step 7. Find member of set S_A for violated constraints (5) [(15)]. Introduce the constraints (5) [(15)] for these members. Set $c:=c+1$. Go to Step 6.

Now we use a very simple example. Suppose the block layout in Figure 11. In this example the following lengths are given:

$$c_{1,2} = c_{3,4} = c_{1,5} = c_{2,6} = c_{3,7} = c_{4,8} = c_{5,6} = c_{7,8} = c_{10,11} = 5,$$

$$c_{2,3} = c_{5,9} = c_{7,10} = c_{8,11} = 10,$$

$$c_{9,10} = 15.$$

Now, we start using the algorithm. This block layout does not have a degree 4 node, so after stage 1 we should go to Step 2. After solving a model with constraints (3), (4) and (7) the following results will be taken:

$$x_{0,7} = x_{0,8} = x_{1,2} = x_{1,5} = x_{2,6} = x_{5,6} = x_{7,8} = 1$$

Figure 12 shows the corresponding network of the answer. The infeasibility of the solution is clear. In stage 4, we should form set S_{A_1} .

$$S_{A_1} = \{1\} \Rightarrow B(S_{A_1}) = \{(1,2), (1,5), (2,6), (5,6)\}$$

Now based on set S_{A_1} , we must add the following constraints because of eliminating the tour:

$$x_{1,2} + x_{1,5} + x_{2,6} + x_{5,6} \leq 3$$

After going back to Step 3 and solving new model, we have:

$$x_{0,5} = x_{0,6} = x_{3,4} = x_{3,7} = x_{4,8} = x_{5,6} = x_{7,8} = 1$$

as the solution. Figure 13 shows the solution. Again we need to add another constraint.

In stage 4, we find:

$$S_{A_2} = \{3\} \Rightarrow B(S_{A_2}) = \{(3,4), (3,7), (4,8), (7,8)\}$$

Therefore, the corresponding constraints will be:

$$x_{3,4} + x_{3,7} + x_{4,8} + x_{7,8} \leq 3$$

In stage 3, again we solve the model. This model will lead to this answer which its configuration is shown in Figure 13:

$$x_{0,6} = x_{0,8} = x_{2,3} = x_{2,6} = x_{3,7} = x_{7,8} = 1$$

Thus, after this step we should stop. The algorithm came to its end and the optimal feasible solution of this problem was taken.

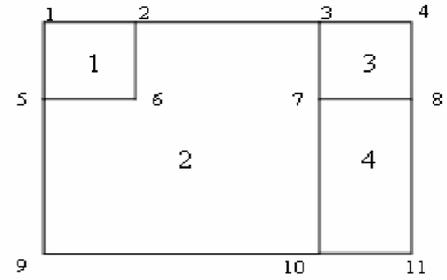


Figure 11. The example.

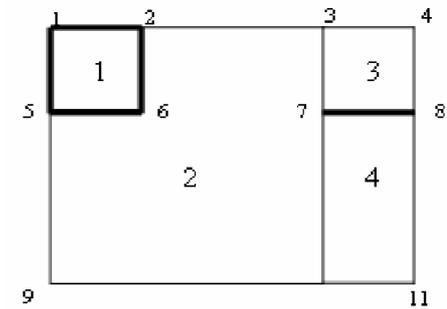


Figure 12. The solution of the first cut.

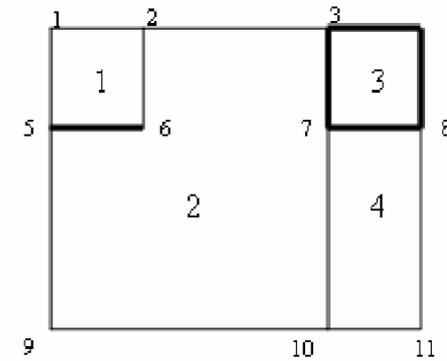


Figure 13. The solution after second cut.

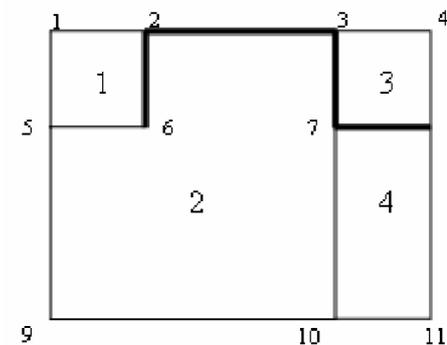


Figure 14. The optimal feasible solution.

5. Computational results

Both algorithms were run on a PC (Pentium 3, 1GHz and 256 MB of RAM). The instances for each run were generated randomly in size of 10, 15, 20, 25, 30, 35, 40 and 45. Also 7 samples for each size were used for getting the computational results which is mentioned in Table 1.

The samples were generated in a randomly manner, because there aren't any standard test problems for this type of problem.

We considered degree 2 nodes in our test problems which result in having much more variables and constraints.

Despite this consideration the computational results show the effectiveness of our algorithms rather than Asef-vaziri et al. [3].

6. Conclusion

The authors have considered the shortest path design problem in this article. Two algorithms were developed for this purpose. In the first algorithm we used a special characteristic of tree to formulate the problem. In each tree, the number of edges is one edge less than the number of nodes. In the second algorithm, we used the approach which was used by Asef-vaziri et al. [3] to solve the SLDP.

With a little change, the authors could use their formulation in our problem. Both algorithms used a simple Branch-and-Cut approach to solve the problem. Using LINGO 8.00 as a commercial ILP solver led to computational results which show that the problem with size of $n \leq 45$ can be easily solved by these algorithms.

Table 1. Results of test problems.

n	Number of solved test problem	1			2		
		Time (second)	#Cut*	#Constraint	Time (second)	#Cut*	#Constraint
10	7	1.89	1.14	1.29	2.71	0.29	0.43
15	7	8.57	1.14	1.29	9.36	1.86	1.86
20	7	38.93	1.14	2.29	37.21	1.43	3.00
25	7	93.64	1.71	2.86	96.57	1.57	2.57
30	7	331.67	2.29	4.00	345.93	2.86	4.71
35	7	1198.74	3.14	5.43	1306.83	3.29	6.43
40	7	1655.29	2.67	5.33	1559.86	3.17	6.67
45	7	6138.44	4.00	7.00	5131.55	5.33	10.17

* In this column, the number of violated answers of the model is indicated which needs to be added some constraints to the model and be solved again.

References

- [1] Afentakis, P., 1989, A loop layout design problem for flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems*, 1, 175-196.
- [2] Apple J. M., 1997, *Plant Layout and Material Handling*. John Wiley & Sons, New York.
- [3] Asef-Vaziri, A., Laporte, G. and Sriskandarah, C., 2000, The block layout shortest loop design problem. *IIE Transactions*, 32, 724-734.
- [4] Asef-Vaziri, A., Dessouky, M. and Sriskandarah, C., 2001, A loop material flow design for automated guided vehicles, *IIE Transactions*, 13, 33-48.
- [5] Bozer, Y. A. and Srinivasan, M. M., 1991, Tandem configuration for automated guided vehicle systems and the analysis of single vehicle loops. *IIE Transactions*, 23(1), 72-82.
- [6] Bozer, Y. A. and Srinivasan, M. M., 1989, Tandem configuration for automated guided vehicle systems offers simplicity and flexibility. *Industrial Engineering*, 21, 23-27.
- [7] Bozer, Y. A. and Srinivasan, M. M., 1995, Tandem AGV systems: A partitioning algorithm and performance. *European Journal of Operational Research*, 63, 173-191.
- [8] Chhajed, D., Montreuil, B. and Lowe, T., 1992, Flow network design for manufacturing system layout. *European Journal of Operational Research*, 57, 145-161.
- [9] De Guzman, M. C., Prabhu, N. and Tanchoco, J. M. A., 1997, Complexity of the AGV shortest path and single-loop guide path layout problem. *International Journal of Production Research*, 35, 2083-2092.
- [10] Egbelu, P. J. and Tanchoco, J. M. A., 1986, Potentials for bi-directional guide-path for automated guided vehicle based systems. *International Journal of Production Research*, 24, 1057-1097.
- [11] Farahani, R. Z., Laporte, G. and Sharifyazdi, M., 2005, An exact algorithm for the shortest loop design problem in a block layout. *International Journal of Production Research*, 43(9), 1879-1887.
- [12] Fischetti, M., Salazar and J. J., Toth, P., 1997, A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research*, 45, 378-394.
- [13] Gaskin, R. J. and Tanchoco, J. M. A., 1987, Flow path design for automated guided vehicle system. *International Journal of Production Research*, 25(5), 667-676.
- [14] Hodgson, T., King, R. and Monteith, S., 1987, Developing control rules for an AGVS using Markov decision processes. *Material Flow*, 4(1), 85-96.
- [15] Kaspi, M. and Tanchoco, J. M. A., 1990, Flow path design for automated guided vehicle systems. *International Journal of Production Research*, 28, 1023-1030.
- [16] Kim, C. W. and Tanchoco, J. M. A., 1991, Conflict-free shortest-time bi-directional AGV routing. *International Journal of Production Research*, 29, 2377-2391.
- [17] Laporte, G., Asef-Vaziri, A. and Sriskandarah, C., 1996, Some application of the generalized traveling salesman problem. *International Journal of Operational Research Society*, 47, 1461-1467.
- [18] Laporte, G. and Nobert, Y., 1983, Generalized traveling salesman problem through n sets nodes: An integer programming approach. *INFOR*, 21, 61-75.
- [19] Maxwell, W. L. and Muckstadt, J. A., 1982, Design of automated guided vehicle systems. *IIE Transactions*, 14(2), 114-124.
- [20] Rajagopalan, S., Heragu, S. S. and Taylor, G. D., 2004, A lagrangian relaxation approach to solving the integrated pick-up/drop-off point and AGV flow-path design problem. *Applied Mathematical Modeling*, 28, 735-750.
- [21] Roe, A., 1997, *User's Guide for LINDO and LINGO, Windows Version*. Duxbury Press.
- [22] Seo, Y. and Egbelu, P. J., 1995, Flexible guide-path design for automated guided vehicle systems. *International Journal of Production Research*, 33, 1135-1156.
- [23] Sinriech, D. and Tanchoco, J. M. A., 1992, The centroid projection method for locating pick-up and delivery stations in a single loop AGV system. *Journal of Manufacturing Systems*, 11, 297-307.
- [24] Sinriech, D. and Tanchoco, J. M. A., 1993, Solution methods for the mathematical models and single-loop AGV systems. *International Journal of Production Research*, 31(3), 705-725.
- [25] Sinriech, D. and Tanchoco, J. M. A., 1994, *SFT-Segmented Flow Topology*, In *Material Flow System in Manufacturing*. Chapter 8, Tanchoco J. M. A. (ed.), Chapman and Hall, London, 200-235.
- [26] Sinriech, D. and Tanchoco, J. M. A., 1995, An introduction to the segmented flow approach for discrete material flow systems. *International Journal of Production Research*, 33,

3381-3410.

- [27] Sinriech, D. and Tanchoco, J. M. A., 1997, Design procedure and implementation of the segmented flow topology (SFT) for discrete material flow system. *IIE Transactions*, 29, 323-335.
- [28] Tanchoco, J. M. A. and Sinriech, D., 1992, OSL-optimal single-loop guide paths for AGVs. *International Journal of Production Research*, 30(3), 665-681.
- [29] Tompkins, J. A., White, J. A., Bozer, Y. A. and Tanchoco, J. M. A., 2002, *Facilities Planning*, 3rd edition, John Wiley & Sons Inc., New York.