A Loop Material Flow System Design for Manufacturing Plants

by

Ardavan Asef-Vaziri

A Thesis Submitted in Conformity with the Requirements for the Degree of Doctor of Philosophy
Graduate Department of Mechanical and Industrial Engineering
University of Toronto

©Copyright by Ardavan Asef-Vaziri

April 3, 1997
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

0-612-27871-9
Abstract

Around 20 to 50 percent of the operating expenses within manufacturing plants are related to material handling. Effective facilities planning can reduce these costs by 10 to 30 percent. The material flow system is the foundation of facilities planning. It is defined in terms of the type of the material handling equipment, the flow network, its directions, and the number and locations of Pick-up and Delivery stations. Automated Guided Vehicles are now considered as the basic material handling equipment in manufacturing plants. They are preferred to conveyors because of their flexibility, and to robots because of their mobility.

We develop a model to design a unidirectional loop flow system for automated guided vehicles. A set of assumptions are examined. Linear Programming, Integer Programming, and Network Flows are modeling tools of our research.

The shortest loop problem is transferred into the Generalized Traveling Salesman Problem. Compared to available models for designing a loop on a block layout, our model is better integrated, has a smaller number of binary variables and stronger constraints. A Goal Programming approach to design the shortest loop with a secondary objective function is also examined.

The shortest loop model minimizes all costs proportional to the length of the loop, but does not cover the travel costs. We integrate both the design of the unidirectional loop, and determination of the number and location of stations in order to minimize the total loaded vehicle travel distance. No earlier model designs the loop and the station locations simultaneously. To solve the problem, we have concentrated on 3 directions. Having a stronger LP-relaxation, an intelligent Branch and Bound, and a set of MIP solver methods which are most appropriate for this particular problem. The model is extended to minimize the cost of the total system, including the network, stations, and loaded travel distance. Capabilities of the model to cover the conventional configuration and the bi-directional loop are also discussed.

Finally the model evaluates the impact of the empty vehicle travel. All considerations of the loaded travel distance formulation and solution procedure, as well as a decomposition approach are implemented to solve the model. We show that the optimal solution for loaded travel distance is far from optimal for both loaded and empty travels.

The models developed in this research may be examined in three environments: fixed loop-variable station location, variable loop-fixed station location, and variable loop-variable station location. However, it is mostly implemented in the last environment. Computational results are reported for a set of randomly generated problems as well as some well known examples.
Acknowledgment

My supervisor, Professor Chelliah Sriskandarajah has technically, spiritually, and financially supported me during the period of my research at U of T. Professor Michael Carter continuously contributed towards the progress of my thesis specially in the last months of my work. I should deeply appreciate the creative participation of these gentlemen in my reset.

I should also thank Prof. G. Laporte, prof. S. Dutta, Prof. J. Abrham, Prof. J. Rogers, Prof. M. Sinclair, Prof. M. Molloy, Prof. D. Krass, my classmates Mr. T. Ganesharajah, and Mr. K. Eshghi for their comments on parts of the thesis.

Completion of this work was impossible if I did not have the spiritual support of Houtan, Heeva, Sara, and Hedih, my four cute and lovely little nieces and nephews.
# Contents

1 Introduction  
1.1 Automated Guided Vehicle Systems ........................................... 2  
1.2 Motivation .................................................................................. 6  
1.3 Perspective of the thesis  
1.3.1 Chapter 2 .............................................................................. 8  
1.3.2 Chapter 3 .............................................................................. 8  
1.3.3 Chapter 4 .............................................................................. 10  
1.3.4 Chapter 5 .............................................................................. 12  

2 Literature Review  
2.1 The Facility Layout Problem ....................................................... 14  
2.2 AGVs Material Flow System Design ............................................ 18  
2.2.1 Loaded Vehicle Travels - Fixed Station Locations - Variable  
Network .......................................................................................... 23  
2.2.2 Empty Vehicle Considerations - Fixed Station Locations - Varia-  
table Network .............................................................................. 26  
2.2.3 Loaded Vehicle Travel - Fixed Network - Variable P/D stations  
2.2.4 Loaded Vehicle Travel - Variable Network - Variable P/D stations  
2.2.5 A Cost Based Models ............................................................. 32  

3 The Shortest Loop for Non-trip Based Material Handling Systems  
3.1 Introduction .................................................................................. 36  
3.2 The Problem  
3.2.1 Formal Statement .................................................................... 37  
3.2.2 The Dual Representation .......................................................... 38  
3.3 Transformation to the Generalized Traveling Salesman Problem  
3.3.1 The Generalized Traveling Salesman Problem  
3.3.2 The Transformation ................................................................. 42  
3.4 Complexity  
3.4.1 Existence of a Feasible Loop ...................................................... 45  
3.5 Lower Bounds  
3.5.1 The Number of Edges ............................................................... 45
### 5 The Optimal Loop Flow System for Empty and Loaded AGV Travel

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction and Review</td>
<td>123</td>
</tr>
<tr>
<td>5.2 Formal Statement</td>
<td>125</td>
</tr>
<tr>
<td>5.3 The Optimization Model</td>
<td>126</td>
</tr>
<tr>
<td>5.3.1 Formulation</td>
<td>126</td>
</tr>
<tr>
<td>5.3.2 Terminology</td>
<td>127</td>
</tr>
<tr>
<td>5.3.3 Description</td>
<td>128</td>
</tr>
<tr>
<td>5.3.4 Empty Vehicle Travel</td>
<td>132</td>
</tr>
<tr>
<td>5.3.5 Additional Constraints</td>
<td>133</td>
</tr>
<tr>
<td>5.3.6 An Upper Bound</td>
<td>133</td>
</tr>
<tr>
<td>5.4 Solution</td>
<td>135</td>
</tr>
<tr>
<td>5.4.1 Decomposition</td>
<td>135</td>
</tr>
<tr>
<td>5.4.2 The Brute Force Search</td>
<td>138</td>
</tr>
<tr>
<td>5.5 Scope for future research</td>
<td>140</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A The Potential Sub-tour Generation Process</td>
<td>154</td>
</tr>
<tr>
<td>B End Points of the Sub-tour Generation Process</td>
<td>157</td>
</tr>
<tr>
<td>C Checking Primitivity and Expanding Procedure</td>
<td>159</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Combinations of the assumptions in the Thesis .......................... 21
2.2 Classification of models .................................................. 22

3.1 The adjacency matrix of our 13-cell example .......................... 58
3.2 The number of potential and primitive sub-tours in a block layout . 70
3.3 Sub-tour elimination constraints required for the 11-cell example 75
3.4 CPU times, number of iterations, and nodes for shortest loop problem
of layouts of sizes 10-40 cells ............................................. 78

4.1 FT-chart for the example of Figure 1 ...................................... 85
4.2 Computation times for randomly generated flows using different solu-
tion procedures ................................................................. 111

5.1 Combinations of the assumptions in this chapter ....................... 125
# List of Figures

1.1 Relationship between throughput and number of vehicles ................................ ..... 7  
2.1 Basic elements of the Material Flow System .................................................. 20  
2.2 Four well known AGVs flow pattern ............................................................... 21  
3.1 Primal and Dual graphs of the 13-cell prototype example ................................ 39  
3.2 Dual graph representation of loop constraints ............................................... 41  
3.3 Transformation required for adjacent nodes of degree 4 ................................ 43  
3.4 An example of a block layout with no feasible loop ....................................... 46  
3.5 Two examples of the class of the block layouts with $\lfloor |C|/2 \rfloor$ as the lower bound for the number of edges on a feasible loop ........................................... 47  
3.6 Two examples of the class of the block layouts with $|C| - |N_k| - 1$ as the lower bound for the number of edges on a feasible loop .................................................. 48  
3.7 An example of a loop with a modified degree 4 nodes ................................... 52  
3.8 Examples of sub-tour and feasible loop ......................................................... 54  
3.9 Representation of a generation of sub-tours .................................................. 57  
3.10 A feasible loop containing 3 cells in the 11-cell example ................................ 60  
3.11 Illustration of main characteristics of sub-tours .......................................... 63  
3.12 Representation of a sub-tour elimination constraint on the primal and dual ........... 65  
3.13 Expanding procedure for sub-tour elimination constraint ............................... 67  
3.14 Expanding a largest wheel to a segment of it .............................................. 69  
3.15 The average number of up to $k$-cell sub-tours in 40 cell layouts ................. 71  
3.16 Horizontal and Vertical CUTS for the 11-cells example ................................ 73  
3.17 Examples of sub-tours which are not eliminated by Horizontal / Vertical cuts ........ 74  
3.18 A CUT constraint eliminates a set of sub-tours ........................................... 76  
3.19 Multiple optimal solutions of the example ................................................ 81  
4.1 The layout of the example from Sinriech and Tanchoco [1993] ...................... 85  
4.2 Graphical representation of the procedure to screen unnecessary primitive sub-tours. ................................................................. 95  
4.3 Optimal solution for loaded vehicles travel ............................................... 109  
4.4 Optimal solutions for multi-stations per cell ............................................. 113
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5 Optimal solution for the cost objective functions</td>
<td>115</td>
</tr>
<tr>
<td>4.6 Relationship between CPU time and cost related functions of the system</td>
<td>117</td>
</tr>
<tr>
<td>4.7 Two examples of special configurations</td>
<td>118</td>
</tr>
<tr>
<td>4.8 Optimal solutions for bi-directional loop</td>
<td>119</td>
</tr>
<tr>
<td>4.9 Optimal solution for Conventional Configuration</td>
<td>121</td>
</tr>
<tr>
<td>5.1 The layout of the example from Tanchoco and Sinriech [1992]</td>
<td>125</td>
</tr>
<tr>
<td>5.2 The first upper bound and the optimal solution for loaded/empty vehicle in the 11-cell example</td>
<td>134</td>
</tr>
<tr>
<td>5.3 The second upper bound for loaded/empty vehicle in the 11-cell example</td>
<td>135</td>
</tr>
<tr>
<td>5.4 Loops with 100 &lt; length &lt; 118</td>
<td>139</td>
</tr>
<tr>
<td>5.5 The last upper bound for loaded/empty travel in 11-cell example</td>
<td>140</td>
</tr>
<tr>
<td>A.1 Potential Sub-Tour Identification Algorithm 1</td>
<td>155</td>
</tr>
<tr>
<td>A.2 Potential Sub-Tour Identification Algorithm 2</td>
<td>156</td>
</tr>
</tbody>
</table>
Terminology

Graphs

\( G_b(N, A) \) : The nondirected graph associated with the block layout.

\( G(C, A) \) : The nondirected adjacency graph corresponding to the block layout.

Two cells are considered adjacent if they have one common edge in the block layout.

\( G_x(C, A_x) \) : The same as \( G(C, A) \) where two cells are considered adjacent if they have one edge or node in common on the block layout.

\( G_o(C_o, A) \) : Adjacency graph of the block layout excluding the external cell.

\( G_f(C, A_f) \) : The flow graph in which for each pair of cells \( c_k \) with a positive flow \( f_{ck} > 0 \) there exist an arc in \( A_f \).

Sets

- \( C \) : Set of cells including the external cell.
- \( C_o \) : Set of cells excluding the external cell.
- \( N \) : Set of nodes.
- \( A \) : Set of undirected edges.
- \( A^\prime \) : Set of directed arcs.
- \( A_x \) : Edges of the graph \( G_x \).
- \( A_{ij} \) : The set of arcs on the directed loop from node \( i \) to \( j \).
- \( N_c \) : Set of nodes on cell \( c \).
- \( \mathcal{V}_n \) : Set of nondirected edges incident to node \( n \).
- \( \mathcal{V}_c \) : Set of nondirected edges forming cell \( c \).
- \( C_n \) : Set of cells adjacent to node \( n \).
- \( \mathcal{P}_c \) : Set of candidate nodes for \( P \) station location on cell \( c \).
- \( \mathcal{D}_c \) : Set of candidate nodes for \( D \) station location on cell \( c \).
- \( N_c \) : Set of intersection nodes on cell \( c \).
- \( \mathcal{X}_c \) : Set of arcs on cell \( c \).
- \( S \) : Set of primitive sub-tours.
- \( R_s \) : Set of nodes on the cells forming sub-tour \( s \).
- \( \overline{R} \) : Complement of \( R \) with respect to \( N \).
- \( NL_{ij} \) : Set of nodes on the line segment of the directed loop from node \( i \) to node \( j \).
$CL_{kij}$ : Set of cells having a non-zero flow with cell $k$, and their pick-up station located on the path $ij$ on the directed loop.

$KL_{cij}$ : Set of destinations of cell $c$ with their $D$ station on the directed loop from node $i$ to $j$.

$N_k^L$ : Set of degree $k$ nodes on a loop.

$N_k^P$ : Set of degree $k$ nodes on a path.

$N_k^T$ : Set of degree $k$ nodes on a tree.

**Parameters**

$l_{mn}$ : Length of edge or arc $mn$.

$f_{ck}$ : Intensity of flow from cell $c$ to cell $k$. The notation $f_c$ is used as the total outflow of cell $c$.

$Q^t$ : The cost of one unit distance of loaded vehicle travel.

$Q_{mn}^e$ : The annuity of the construction cost, space occupied, maintenance, and other costs associated with arc $mn$ on the loop.

$Q_{cn}^e$ : The annuity of the construction cost, space occupied, maintenance, and other costs associated with each station $n$ on cell $c$.

$US_n$ : Maximum number of stations at node $n$.

$UF_n$ : Maximum flow transferred through stations at node $n$.

**Decision Variables**

$X_{mn}$ : Binary decision variable corresponding to directed arc $mn$. The variable is equal to one if the arc is on the loop, and zero otherwise.

$Y_{mn}$ : Binary decision variable corresponding to non-directed edge connecting nodes $m$ and $n$, where $n > m$. The variable $Y_{nm}$, where $n > m$ is only an alias name for $Y_{nm}$.

The variable is equal to one if the edge is on the loop, and zero otherwise.

$P_{cn}$ : Binary decision variable corresponding to node $n$ as Pick-up station of cell $c$, which is one if node $n$ is selected and zero otherwise.

$D_{cn}$ : Binary decision variable corresponding to node $n$ as Delivery station of cell $c$ which is one if node $n$ is selected, and zero otherwise.

$p_{c kn}$ : Binary decision variable corresponding to node $n$ as $P$ station of flow $ck$,
which is one if selected, and zero otherwise.

d_{kn} \quad \text{Binary decision variable corresponding to node } n \text{ as } D \text{ station of flow } ck, \text{ which is one if selected, and zero otherwise.}

t_{ckmn} \quad \text{Intensity of flow from cell } c \text{ to cell } k \text{ passing through arc } mn.

\text{The notation } t_{cmn} \text{ is used as the intensity of the total outflow of cell } c \text{ passing through arc } mn.

L_{ij} \quad \text{Length of the line segment on the directed loop from node } i \text{ to node } j.

O_c^{(i)} \quad \text{Contribution of origin cell } c \text{ in total outflow travel in the system when node } i \text{ is selected as its } P \text{ station.}

O_c \quad \text{Contribution of cell } c \text{ in total outflow travel in the system.}

I_k^{(i)} \quad \text{Contribution of destination cell } k \text{ in total inflow travel in the system resulting from selection of node } i \text{ as its } D \text{ station.}

I_k \quad \text{Contribution of cell } k \text{ in total inflow travel in the system.}

\omega_{mn} \quad \text{Total flow passed through arc } mn.

\Lambda_{ck} \quad \text{The set of arcs on the loop from } P \text{ station of cell } c \text{ to } D \text{ station of cell } k.

\Theta_\Gamma \quad \text{The maximum flow passed through an arc of the cycle } \Gamma.
Chapter 1

Introduction

One of the oldest activities of industrial engineering is facilities layout, which deals with the design of an arrangement of physical elements of manufacturing systems. The drawing of the resulting design is known as the Plant Layout. A good layout always involves the methods of handling material; therefore *Plant Layout and Material Handling* are two cohesively integrated terms.

The overall objective in facility layout and material handling is an effective and efficient arrangement of required facilities to transfer inputs into outputs. In the manufacturing context, machines or equipment, work stations, production cells or departments, and auxiliary services are among facilities, Apple [1977]. It has been estimated that between 20 to 50 percent of the total operating expenses within manufacturing plants can be attributed to material handling. Furthermore it is generally agreed that effective facilities layout can reduce these costs by 10 to 30 percent, Tompkins, White, Bozer, Frazelle, Tanchoco, Trevino [1996]. Facilities layout is a long term, costly operation, and any unplanned modification or rearrangement results in a large expense. Facilities layout design is divided into Basic Design and Detail Design phases. The major potential gains are perceived to be associated with the basic design phase. Montreuil [1990].

For more than 30 years, the block layout was considered to be the final outcome of the Basic Layout Design Phase. The block layout is a drawing partitioned into
a number of right angle polygons. Each polygon corresponds to a machine, work center, cell, department, shop, etc.\textsuperscript{1} It shows the relative positioning of cells, based on a hypothetical free rectilinear flow of material, rather than taking into account actual flow through a network of aisles.

Today it is believed that the block layout should be integrated with the material flow system. The material flow system is defined in terms of the type of the material handling equipment, the flow system, its directions, and the number and locations of pick-up (input) and delivery (output) stations, Montreuil [1990].

Automated Guided Vehicles are now considered to be the final mode of material handling in manufacturing plants. They are preferred to robots due to their mobility, and to conveyors due to their flexibility.

In this chapter we will have an introduction on Automated Guided Vehicle Systems. Then the motivations for our research are discussed. Finally, the content of the thesis is overviewed.

1.1 Automated Guided Vehicle Systems

Automated Guided Vehicle Systems (AGVS) have been of great interest to industry for the past 20 years. They are also an important component in Computer Integrated Manufacturing. The applications of Automated Guided Vehicles have increased to a point where AGVS are considered to be a basic concept in material handling. Although initial applications of AGVS were generally limited to warehouses, their utilization in manufacturing systems has been rapidly increasing. They play an important role in many low to medium volume manufacturing operations, including Flexible Manufacturing Systems, Kusiak [1985].

AGVs are driverless electronically guided vehicles capable of loading, transporting, and unloading without human intervention. Vehicles are independently addressable

\textsuperscript{1}In the remainder of this thesis, the term cell is used as the common name for Department, Division, Center, shop, Work station, Facility, Machine, Activity, etc.
and centrally controlled. Push carts and forklifts are inferior to AGVs because of space utilization, possibility of error, safety considerations, and overall operating costs. Although AGVs are significantly more expensive than forklifts or other manual handling equipment, they pay for themselves rapidly by reducing unit operating costs, and multiple shift operations. Robots can not provide the mobility of the automated guided vehicle system, and conveyors do not offer the flexibility. AGVS do not create physical barriers within the factory as conveyors do. Therefore, they can share space with other uses such as pedestrian or forklift aisles, improving overall space utilization.

Vehicles move along a transportation network appropriate to the specific application of the AGVS. The flow network can be easily modified. Benefits of AGVS include better inventory control, higher utilization, productivity, flexibility, and easier operations management.

The first large scale manufacturing application of an AGV system occurred in 1974 at a Volvo plant in Kalmar, Sweden. In less than 10 years, about 3,300 plants worldwide employed more than 15,000 AGVs, Zygmont [1986]. The largest application in North America is at the truck assembly plant of General Motors in Oshawa, Ontario, Canada. Where there are more than 1000 AGVs transporting truck engines, bodies, and chassis across the 2.7 million square feet plant, Gould [1987]. However, Japan and Europe lead in AGV system applications. For example, in 1989, Japanese companies purchased 5000, European 3000, and US companies 500 vehicles.

There are 6 basic types of AGV; Unit Load, Towing, Pallet Truck, Fork Truck, Light Load, and assembly line vehicles. We are concerned with the Unit Load vehicles. Designing optimal unit loads is a duty of facility planners. Large unit loads result in the need for fewer AGVs, but require sophisticated handling, and more clamping devices and fixtures. Small unit loads require more vehicles, but less expensive ones.

AGVS have 5 basic functions; Guidance, Routing, collision avoidance, Load Transfer, and System Management.

*Guidance* systems are used to determine vehicle paths, and to implement phys-
A Loop Material Flow System Design for Manufacturing Plants

A loop material flow system allows the desired transportation network. There are two types of guidance system; those using off-board fixed paths, and those using on-board software programmed paths. Wire guided, where vehicles follow a wire buried in the floor, is the earliest off-board types. In this system, *The Steering Control* allows the vehicles to maneuver in different ways. There are two basic types of steering control; *Differential Speed Control* uses two fixed wheel drives and varies the speed between the two drives on either side of the guide path. *The Steered Wheel Control* uses a front steered wheel to turn and follow the guide path. The painted line guided, where vehicles follow a fluorescent line by sensing its reflections, is another example of off-board fixed paths. On-board software programmed paths or virtual flow paths allows vehicle movements without a physical guide path.

*Routing* is the decision making ability to follow a route along the guide path. There are two methods of routing in wire guidance. In *The Frequency Select*, when the vehicle reaches the decision point, two frequencies are present in the same slot. The vehicle selects the frequency corresponding to the direction it wishes to go, and the routing is automatically accomplished. In *The Path Switch Select Method*, the vehicle approaches the decision point and passes an activation device which leaves only one path open.

*Collision Avoidance*. There are two types of collision avoidance methods. In *Zone Control*, the path is segmented into a set of zones, only one vehicle is permitted in a zone. The vehicle must proceed to the next zone before the trailing vehicle can move in. Blocking occurs when a vehicle intend to enter into a zone, but the zone is already occupied by another vehicle. Blocking is a critical issue in design and operations of AGVS. The higher the number of zones, the lower the blocking, but the higher the cost of zoning. Collision avoidance by forward sensing uses an on board sonic, optical, or bumper method to detect the vehicle in front.

*Load Transfer* is the capability of load pick-up and Drop-off. There are Manual, Conveyor, Lift/Lower, Push/Pull methods of load transfer. The unit load AGVs
can be equipped with conveyor transfer decks used to automatically transfer loads between the vehicle and the P/D station. In automatic transfer of loads, a method of hand shaking is employed, by which the vehicle checks to make sure that there is a load. The hand shake signal activates the load transfer device, and also stops it when the transfer is complete, Koff and Boldrin [1985].

*System management* includes; Dispatching, Vehicle Routing, Traffic Management, and Monitoring.

*Dispatching* rules are a set of procedures to assign a vehicle to a work station. They are divided into two common classes; Work Center initiated, and Vehicle initiated. Some Vehicle/Station assignment rules are; random, shortest/longest travel time or distance, length of queue/remaining queue, modifications of first come first serve, etc. In most of the dispatching rules, a vehicle after each trip remains at the delivery station until another trip is assigned to it. Upon receiving the message from the central monitoring system, the vehicle moves to the next assigned pick-up station. In some dispatching rules, after each trip, the vehicle is sent to a central terminal, in some others it is returned to the P station it comes from. In another, the vehicle is always traveling and checking the upcoming pick-up stations for a load.

*Vehicle routing* is the methodology to select a route for each trip. The criteria for route selection is to minimize the travel time which is the integrated impact of travel distance and zone blocking.

*Traffic management* is the procedure to assign priority to vehicles to pass an intersection or enter a zone. An important part of AGVS is the information system that handles the transfer of information between the central computer and peripheral controllers on-board the vehicles.

Industrial applications and the state of the art in AGV systems are regularly reviewed in trade publications such as *Modern Materials Handling* and *Material Handling Engineering*. Gould [1988, 1990b], Schwind [1988], and Forger [1996]. For mechanical design of AGVs and their guidance methods the reader is refereed to Koff.
[1987], Vosniakos and Mamalis [1990]. The April 1996 issue of IIE Solutions, contains a buyer’s guide for Automated Guided Vehicles. The guide covers vehicle types, guidance methods, communication systems, load capacities, travel speeds, battery types, and price ranges.

1.2 Motivation

From the Industrial Engineering and Management Science point of view, the most important issue in AGVS is the material flow system design. The material flow system design in its general version is to find a sub-set of arcs among a set of candidate edges in the block layout to be included in the flow pattern, the directions of arcs, the number of pick-up and delivery stations at each cell, the location of each P/D station, and finally an improved layout of production cells, Tanchoco and Sinriech [1992]. The objectives of the material flow system are; the minimization of the total length of flow pattern, the total loaded vehicle travel, the total loaded and empty vehicle travel, and the total costs of flow path, P/D stations, and vehicle travel.

Analysis of material flow systems is comprised of two stages; design and operations. In operations the static nature of the design turns into its dynamic reality. The analytical solution for material movement in a complicated network naturally is better than that of a simple configuration. However, due to blocking, the situation may not remain the same in the real operations. Indeed all 3 duties of dispatching, vehicle routing, and traffic management are substantially simplified when the flow system is simplified. These are finally translated into less blocking and less expensive software requirements.

In addition, as a result of excessive blocking in complicated networks, the expected fleet size is not capable to transfer the required volume of material through the flow system. To meet the required throughput, more AGVs are added to the system, which in turn, create more blocking. It is quite possible that the manufacturing system never achieves its designed throughput. Figure 1.1 is a hypothetical representation of such
a situation. Clear understanding of blocking consequences is an essential issue in flow path design. It may lead to a simple and flexible configuration which could result in both a smaller fleet size, and also higher throughput.

The simplicity of the flow network, while maintaining a high degree of efficiency and flexibility, is of special consideration in designing material flow systems. Four simple general flow patterns are identified as Straight line, Zigzag, U-shaped, and Loop, Apple [1977]. Loop configuration because of its efficiency and simplicity represent an attractive solution to the manufacturing system layouts. It is attractive from the point of initial investment, product and processing flexibility, and expansion costs. Afentakis [1989] proposed a loop material flow pattern for flexible manufacturing systems. Bartholdi and Platzman [1989], suggested a single loop for Automated Guided Vehicle Systems. Kouvelis and Kim [1992], designed a unidirectional loop layout for automated manufacturing systems.

Flow networks can be defined as being either unidirectional or bi-directional. In unidirectional flow, vehicles are restricted to travel only in one direction along a given
section or on the whole flow network. Unidirectional flow paths have been extensively implemented since they reduce the probability of vehicle blocking. In this thesis, we design a unidirectional loop flow system.

1.3 Perspective of the thesis

In this section we summarize the content of each chapter of the thesis.

1.3.1 Chapter 2

In chapter 2 we review and critique the available literature on material flow system design. The models are classified into the environments of. 1) Loaded vehicle travel objective, variable network, fixed station locations. 2) Empty vehicle considerations, variable network, fixed station location. 3) Loaded vehicle travel, fixed network, variable station location. 4) Loaded vehicle travel, variable network, variable station location. 5) Total cost objective function, variable network, variable station location.

1.3.2 Chapter 3

One main objective in designing a material flow pattern is the minimization of the total length of the flow network. The minimization of the length of the network is a proxy for minimizing the initial fixed construction, installation, space occupied, and all other yearly costs proportional to the length of the loop. This objective is of crucial importance in designing non-trip based material handling systems. On ground conveyors like belt and roller, overhead conveyors like trolley, automated monorail, and power and free are among non-trip based material handling systems. In chapter 3 a model is developed to design the shortest loop which covers at least one edge of each cell on a block layout. A Goal Programming approach to design the shortest loop with secondary objective functions is also examined. The secondary objective functions are to minimize the number of edges, to maximize the sum of the weighted
nodes, and to minimize the corner points. These objectives have specific physical interpretation in the layout design.

Our work differs from that of others on the block layout shortest loop problem in several main aspects.

1) A new formulation with a smaller number of binary variables, and a stronger set of constraints is developed.

2) Based on the properties of the structure of the block layout, a different sub-tour elimination approach is implemented. We will show that the following properties differentiate the problem from a general version of Traveling Salesman Problem.

2-1) Since $G(C,A)$, the dual of the graph associated with the block layout is far from complete, the number of potential sub-tours is much less than $2^{|C|-1}$. The special structure of the graph coupled with a new interpretation of a sub-tour tailored for our specific problem are the main reasons for having a smaller number of sub-tours.

2-2) A collection of adjacent cells where one of them does not have an edge on the boundary of the composite shape is an impossible sub-tour.

2-3) Unlike the TSP, sum of the number of cells in a sub-tour $s$, and that of its complement $\bar{s}$ is neither $|C|$ nor constant, $|s| + |\bar{s}| < |C|$. The complement of a sub-tour $s$, is the remainder of the graph where both $s$ and its adjacent cells are removed. Therefore, the steady state criteria of $|s| \leq \lfloor |C|/2 \rfloor$ for a sub-tour to be Primitive, i.e., its own elimination constraint to be included in the model, is replaced by the transient criteria of $|s| \leq |\bar{s}|$.

2-4) The sub-graph induced by $\bar{s}$ is not necessarily connected. Therefore, if $r$ is the smallest component of $\bar{s}$, the elimination constraint of the sub-tour $s$ is required if and only if $|s| \leq |r|$.

2-5) Given the set of primitive sub-tours $S = \{s : |s| \leq |r|\}$. There is a dominant sub-set $S' \subseteq S$, such that given any sub-tour $s \in S \setminus S'$, there exists a sub-tour $s' \in S'$ such that the elimination constraint of $s'$ is at least as strong as that of $s$. Therefore, $S$ shrinks into $S'$. 
3) The shortest loop is found directly. In earlier formulations, the method borrowed for the purpose of sub-tour elimination requires duplication of a node. The duplicated node is assumed as the first and the last point of the travel, it is automatically included in any solution. It does not create any difficulty in the TSP because all nodes are covered by a feasible as well as optimal solution. However, in our problem, it is not known in advance which nodes are covered by the optimal loop. Fixing any node in the final solution immediately cuts a portion of the search space, a portion which may include the optimal solution. Therefore, the formulation must be duplicated for more than one node. Our formulation directly finds the shortest loop.

4) The sub-tour elimination constraint mentioned above has a weak LP-relaxation. There are some ways to make it stronger, but no modification is reported in loop flow models for block layout. The constraint implemented in our formulations has the strongest LP-relaxation among all sub-tour elimination constraints. Due to these features, the computation times in our formulation are much lower. Indeed for small to medium size problems, the optimal solution is found in the root node of Branch and Bound.

1.3.3 Chapter 4

Minimization of the loaded vehicle travel distance is the objective of most of the available research on AGV material flow pattern. In chapter 4 we develop a model to integrate both the design of the unidirectional loop material flow pattern, and determination of the number and location of pick-up and delivery stations. Edges and nodes of a fixed block layout are candidates for the flow path and pick-up/delivery station locations. The objective function is to minimize the loaded vehicle travel distance.

The model contains 3 sets of constraints, a) Loop configuration, b) Station locations, and c) Material flow. Two factors have the most important impact on the
solution time; size of the layout and intensity of From-To chart. The formulation is
examined against a set of well known layouts, and randomly generated flow graphs.
Several sets of quite dense flows were generated from uniform distribution of (0,100).
Half of the F-T Charts are 25 percent filled non-zero elements, the remaining are 50
percent dense.

In improving the effectiveness of the formulation and the efficiency of the solution
procedure, we have concentrated on 3 directions.

1) To replace a weak LP-relaxation by a strong one. This is done by a better
formulation, proliferation of constraints, and preprocessing. It improves the quality
of the solution at the root node.

2) To replace a naive Branch and Bound by an intelligent one. A taxonomy of
variables is defined for the purpose of branch and bound. Its impact on the LP-
relaxation of the problem at subsequent nodes is discussed. The proposed taxonomy
of variables directs the solution procedure to branch first on edges, second on arcs,
then on P stations , and last on D stations. The variables within each of the above
4 categories are also prioritized using some heuristics.

3) To replace the standard set of LP/IP routines with a set being the most efficient
for our specific problem. In order to identify the most efficient combination to solve
the problem, different routines are examined on the root and subsequent nodes. Both
network simplex and dual simplex completely outperformed primal simplex at the
root node. In subsequent nodes, dual dominated primal by a very large gap. The
steepest edge pricing improves solution time in all problems. Back tracking to the
node with best estimated integer objective function when all integer infeasibilities
are removed dominated depth first as well as best linear objective function search
procedure.

The problems were first solved using primal simplex at both root and subsequent
nodes. Average CPU times for 25 and 50 percent dense problems were 2501 and 4795
seconds respectively. These numbers for the best procedure implemented on first and
second sets of problems reduced to 65 and 417 seconds respectively.

The solution times for the case of multiple stations per cell are much lower. That is because, all station variables are moved into real variables. In an extension of the model, all elements of the material flow system are integrated into a cost-based model. The cost elements of the system are cost of the loop, stations, and loaded vehicle travel. The problem is to find a loop with minimum cost for the total system.

The solution time for this model is sensitive to the relative weights of cost parameters. If cost parameters are close to their trade-off, the CPU times increase. Indeed more and more nodes have their lower bound below the current integer solution.

To the best of our knowledge, the only analytical models addressing multiple stations per cell are that of Goetz-Egbelu [1990], and Palliyil and Goetschelckx [1994]. This later, is the only analytical model taking into account the fixed charge of P/D stations.

Extension to conventional configuration, bi-directional loop are also modeled.

1.3.4 Chapter 5

Despite the radical impact of the empty vehicle travel on the material flow system design, to the best of our knowledge, there is no optimal or heuristic procedure considering this issue in a single loop material flow system. In chapter 5 we design the optimal loop to minimize the total empty and loaded travel.

Sinriech and Tanchoco [1992] state that the impact of the empty vehicle travel on the performance of the single loop is negligible. They compare the throughput of a single loop and a conventional configuration under a variety of dispatching rules. It is shown that regardless of the dispatching rule, the single loop performance is much more robust than the conventional configuration. The loop implemented in this comparison is not the optimal solution to their loaded vehicle travel model. The corresponding conventional configuration is also not an optimal solution. We will show that ignoring empty vehicle travel has a substantial impact on the optimal
solution.

In chapter 5 integrated model is developed to design a loop, determine its direction and the location of P/D stations in order to minimize total empty and loaded vehicle travel distance. In all models developed for the minimization of loaded vehicle travel in a single loop, intersections on the block layout are the candidate locations for P/D stations. In order to show the impact of the empty vehicles on the optimal solution of these models, we first solve the problem for the case of stations on intersections. It is shown that the first lower bound for empty/loaded travel in the prototype example is 47 percent better than what is obtained by the optimal solution for loaded vehicle travel.

In order to not end up with an unrealistic optimal solution in which many P/D stations are clustered together, and to avoid blocking, the model is extended to the case of stations on edges.

The model could be examined in three environment: Fixed Loop/Variable Stations, Fixed Stations/Variable Loop, and Variable Loop/Variable Stations. However, the first and second environments are special cases of the last one. Primal/Dual as well as Arc/Path formulations are developed. A decomposed brute force solution procedure is discussed. Computational results are reported for a set of randomly generated layouts and flow graphs, as well as some well known examples extensively used in literature.

All computations are carried out using the general purpose MIP solver CPLEX3.0 on a SUN Sparc station Model 20.
Chapter 2

Literature Review

In this chapter we review and discuss the available literature on Unidirectional Material Flow Systems for Automated Guided Vehicle Systems.

2.1 The Facility Layout Problem

The facility layout problem is mainly modeled as the Quadratic Assignment Problem. QAP is NP-complete, Sahni and Gonzalez [1976]. The largest solved problem reported in the literature is of size 17.

Koopmans and Beckman [1957] were the first who modeled the problem of locating plants with material flow between them. Their QAP formulation along with improvements made by Lawler [1963], is as follow

$$\text{Minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} b_{ijkl} x_{ij} x_{kl}$$ \quad (2.1)

$$s.t. \quad \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, 2, \ldots, n \quad (2.2)$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, 2, \ldots, n \quad (2.3)$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1, 2, \ldots, n \quad (2.4)$$

$b_{ijkl} = a_{ij}$ if $i = k$ and $j = l$, otherwise is equal to $f_{ik}d_{jl}$. The parameter $a_{ij}$ is the fixed cost of locating a facility at a location, $f_{ik}$ is the material flow between two
facilities, \( d_{ij} \) is the distance from one location to another.

To linearize the objective function, Lawler [1963], also replaced \( x_{ij}, x_{kl} \) by \( y_{ijkl} \). The following constraints were added to the problem to regulate the relationship between the binary variables \( x_{ij}, x_{kl} \) and the real variable \( y_{ijkl} \).

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} x_{ij} + x_{kl} - y_{ijkl} \leq 1 \tag{2.5}
\]

\[
x_{ij} + x_{kl} - 2y_{ijkl} \geq 0 \quad i, j, k, l = 1, 2, \ldots, n \tag{2.6}
\]

Bazaraa and Sherali [1980], made the substitution \( g_{ijkl} = \frac{(a_{ij} + a_{kl})/(n - 1)}{+ (f_{ik} + f_{ki})d_{jl}} \). Their model along with constraints (2.2)-(2.4) is as follow:

\[
\text{Minimize} \quad \sum_{i=1}^{n-1} \sum_{j=1}^{n} \sum_{k=i+1}^{n} \sum_{l=1}^{n} g_{ijkl} y_{ijkl} \tag{2.7}
\]

\[
\text{s.t.} \quad \sum_{i=1}^{n} y_{ijkl} - (n - i)x_{ij} = 0 \quad i = 1, 2, \ldots, n - 1 \quad j = 1, 2, \ldots, n \tag{2.8}
\]

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} y_{ijkl} - (k - 1)x_{kl} = 0 \quad k = 2, \ldots, n - 1 \quad l = 1, 2, \ldots, n \tag{2.9}
\]

\[
y_{ijkl} \leq 1 \quad i = 1, 2, \ldots, n - 1 \quad k = i + 1, \ldots, n \quad j, l = 1, 2, \ldots \quad j \neq l \tag{2.10}
\]

Note that \( y_{ij,kj} \) or \( y_{ij,il} \) exists neither in this formulation nor in any of the following formulations. Kaufman and Broecks [1978] implemented the substitutions of the real variables \( w_{ij} = x_{ij} \sum_{k=1}^{n} b_{ijkl}x_{kl} \) and \( e_{ij} = \sum_{k=1}^{n} \sum_{l=1}^{n} b_{ijkl} \). Their formulation which has the smallest number of variables and constraints, along with constraints (2.2)-(2.4) is

\[
\text{Minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \tag{2.11}
\]

\[
e_{ij}x_{ij} + \sum_{k=1}^{n} \sum_{l=1}^{n} b_{ijkl}x_{kl} - w_{ij} \leq e_{ij} \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n. \tag{2.12}
\]

However, the number of constraint in general, and the number of variables in some problems, are not the measure of effectiveness of MIP formulations.

Indeed, as we explain in chapter 4, in many instances the reverse is true.
The Mixed Integer Programming formulation of Love and Wong [1976] implements rectilinear distances. A real variable \( h_{ik}^+ \) is equal to the horizontal distance between facilities \( i \) and \( k \) when facility \( i \) is to the right of facility \( k \), and is 0 otherwise. The real variables \( h_{ik}^-, v_{ik}^+, v_{ik}^- \) define the distances in the three other directions. The real variables \( x_i, y_i \) represent the coordinates of the location of \( a_j, b_j \) which are the coordinates of location \( j \). Constraints (2.2)-(2.4) remain the same.

\[
\text{Minimize } \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}x_{ij} + \sum_{i=1}^{n-1} \sum_{k=i+1}^{n} f_{ik}(h_{ik}^+ + h_{ik}^- + v_{ik}^+ + v_{ik}^-) \quad (2.13)
\]

\[
s.t. \quad x_i - x_k = h_{ik}^+ - h_{ik}^- \quad (2.14)
\]
\[
y_i - y_k = v_{ik}^+ - v_{ik}^- \quad (2.15)
\]
\[
i = 1, 2, \ldots, n - 1 \quad k = i + 1, \ldots, n
\]
\[
x_i + y_i = \sum_{j=1}^{n} (a_j + b_j)x_{ij} \quad i = 1, 2, \ldots, n \quad (2.16)
\]
\[
x_i - y_i = \sum_{j=1}^{n} (a_j - b_j)x_{ij} \quad i = 1, 2, \ldots, n \quad (2.17)
\]

There are several other equivalent Linear Integer Programming Formulations of QAP (Balas and Mazzola [1980], Burkard and Bonninger [1983], Frieze and Yadegar [1983], Heragu and Kusiak [1991].

The distance between pairs of locations in some situations depend upon the facilities assigned to them. One way of modeling this situation is to replace \( c_{ji} \) by \( c_{ij}^k \), which is the unit load transportation cost from location \( j \) to location \( l \) under arrangement \( k \).

The Quadratic Set Covering Problem, [Bazaraa, 1975] is another formulation of this problem. The total area occupied by all facilities is divided into \( q \) blocks.

\[
\text{Minimize } \sum_{i=1}^{n} \sum_{j=1}^{l(i)} a_{ij}x_{ij} + \sum_{i=1}^{n} \sum_{j=1}^{l(i)} \sum_{k=1}^{l(j)} \sum_{l=1}^{l(k)} f_{ik}d_{ijkl}x_{ij}x_{kl} \quad (2.18)
\]

\[
s.t. \quad \sum_{j=1}^{l(i)} x_{ij} = 1 \quad i = 1, 2, \ldots, n \quad (2.19)
\]
\[ \sum_{i=1}^{n} \sum_{j=1}^{I(i)} p_{ijt} x_{ij} \leq 1 \quad t = 1, 2, \ldots, q \]  

\( I(i) \) is the number of potential locations for facility \( i \). \( d_{ijkl} \) is the distance between the centroids of locations \( j \) and \( l \) when assigned to facility \( i \) and \( k \) respectively. \( p_{ijt} \) is 1, if block \( t \) belongs to the set of blocks occupied by facility \( i \) if located at \( j \), and is 0 otherwise.

Since the distance between two locations is the distance between their centroids, the alternative formulation is

\[ \text{Minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{q} a_{ij} x_{ij} + \sum_{i=1}^{n} \sum_{j=1}^{q} \sum_{k=1}^{q} \sum_{l=1}^{q} f_{ik} d'_{ijl} x_{ij} x_{kl} / s_i s_k \]  

\[ s.t. \quad \sum_{j=1}^{q} x_{ij} = s_i \quad i = 1, 2, \ldots, n \]  

\[ \sum_{i=1}^{n} x_{ij} \leq 1 \quad j = 1, 2, \ldots, q \]  

\[ x_{ij} \in \{0, 1\} \quad i = 1, 2, \ldots, n \quad j = 1, 2, \ldots, q \]  

where \( s_i \) is the number of blocks occupied by facility \( i \), and \( d'_{ijl} \) is the distance between blocks \( j \) and \( l \).

Montreuil [1990], developed a modeling framework to integrate block layout, flow path and location of P/D stations, no solution procedure is suggested. The formulation with slight modifications is as follows.

\[ \text{Minimize} \quad \sum_{e \in E} M_{ce} + \sum_{m,n \in E} l_{mn} \sum_{f \in F} t_{mnf} \]  

\[ s.t. \quad L X_e \leq X_e - Y_e \leq U X_e \quad \forall c \in \{C \cup B\} \]  

\[ L Y_e \leq Y_e - X_e \leq U Y_e \quad \forall c \in \{C \cup B\} \]  

\[ P_c \leq 2(X_e - X_e + Y_e - Y_e) \leq P_c \quad \forall c \in \{C \cup B\} \]  

\[ X_B \leq X_e \leq X_B \quad \forall c \in C \]  

\[ Y_B \leq Y_e \leq Y_B \quad \forall c \in C \]
\[ X_c \leq x_{cs} \leq \bar{X}_c \quad \forall c \quad \forall s \in S_c \tag{2.31} \]
\[ Y_c \leq y_{cs} \leq \bar{Y}_c \quad \forall c \quad \forall s \in S_c \tag{2.32} \]
\[ R_{ck} + R_{ck} + R_{ck} + R_{ck} \geq 1 \quad \forall (c < k, c, k \in C) \tag{2.33} \]
\[ R_{ck} + R_{ck} \leq 1 \quad \forall (c < k, c, k \in C) \tag{2.34} \]
\[ R_{ck} + R_{ck} \leq 1 \quad \forall (c < k, c, k \in C) \tag{2.35} \]
\[ I_f A_{D(I),n} + \sum_{m \in E} t_{mn} = I_f A_{D(I),n} + \sum_{m \in E} t_{mn} \quad \forall n \in N \quad \forall f \in F \tag{2.36} \]
\[ \sum_{f \in F} t_{mn} \leq C_{mn} \quad \forall mn \in E \tag{2.37} \]
\[ \sum_{\forall n \in N} A_{csn} = 1 \quad \forall c \in C \quad \forall s \in S_c \tag{2.38} \]
\[ x_{csn}^+ - x_{csn}^- = x_{cs} - x_n \quad \forall c \in C \quad \forall s \in S_c \quad \forall n \in N \tag{2.39} \]
\[ y_{csn}^+ - y_{csn}^- = y_{cs} - y_n \quad \forall c \in C \quad \forall s \in S_c \quad \forall n \in N \tag{2.40} \]
\[ r_{cs} \geq x_{cs}^+ + x_{cs}^- + y_{cs}^+ + y_{cs}^- - M(1 - A_{csn}) \quad \forall c \in C \quad \forall s \in S_c \quad \forall n \in N \tag{2.41} \]

Where \( X_c, \bar{X}_c, Y_c, \bar{Y}_c \) are lower and upper boundaries of cell \( c \) on \( X \) and \( Y \) directions respectively. \( LX_c, \bar{LX}_c, LY_c, \bar{LY}_c, P_c, \bar{P}_c \) are lower and upper bounds on the length, width, and perimeter of cell \( c \) respectively. \( x_{cs}, y_{cs} \) are the coordinates of P/D stations. \( R_{ck} \) is equal to 1 if cell \( c \) is to the left of cell \( k \), and 0 otherwise. \( R_{ck} \) is equal to 1 if cell \( c \) is above cell \( k \), and 0 otherwise. The binary variable \( A_{csn} \) is equal to 1, if station \( s \) of cell \( c \) is anchored to node \( n \).

### 2.2 AGVs Material Flow System Design

Co and Tanchoco [1991] summarize the research on dispatching rules, routing, and scheduling of AGVS. King and Wilson [1991] review the research on flow path design, fleet size, routing, scheduling, and the justification for implementing AGVS.
Johnson and Brandeau [1996] summarize the stochastic models available for fleet size determination and routing.

The elements of AGVs material handling systems are shown in Figure 2.1. We review those papers concerning one or more components of the illustrated total system for the unit load AGVs.

Four well known flow configurations are shown in Figure 2.2. In a Conventional Configuration all edges of the block layout are on the flow network, 2.2(a). A Unidirectional Loop Flow Pattern covers at least one edge of each cell, 2.2(b). A Tandem Configuration partitions the flow pattern into a set of disjoint bi-directional loops. Each loop is served by a single vehicle, 2.2(c). Segmented Flow Topology is a special instance of the tandem configuration. Its network which is not necessarily connected, is partitioned into a set of segments. One material handling equipment operates in each segment, 2.2(d). Flow path design of all above configurations seem intractable. However, we are unaware of any formal proof of NP-Completeness of any configuration.

In this review we mainly concentrate on optimization models and conventional heuristics for designing material flow systems. We do not review implementations of the recent heuristics like Simulated Annealing, Generic Algorithm, Tabu Search.

Automated Guided Vehicle Material Flow Systems were first modeled by Maxwell and Muckstadt [1982]. The model is mainly vehicle routing, however, material flow path and station location issues are also addressed. The unidirectional network and the location of stations are fixed. Unit load AGVs are utilized. The objective is to determine vehicle routes and the number of required vehicles. Blair, Charnsethikul, Vesques [1985], also addressed the problem of vehicle routing where the flow network and the station locations were already designed. The objective function of their IP formulation was to minimize the maximum distance a vehicle travels.

The available literature could be categorized into 5 main environments explained in Table 2.1. Each environment may also be divided into loaded or both loaded and
Figure 2.1: Basic elements of the Material Flow System
Figure 2.2: Four well known AGVs flow pattern

<table>
<thead>
<tr>
<th>Environment</th>
<th>Flow Loop</th>
<th>Station Location</th>
<th>Number of Stations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fixed</td>
<td>Variable</td>
<td>One</td>
</tr>
<tr>
<td>2</td>
<td>Fixed</td>
<td>Variable</td>
<td>Multiple</td>
</tr>
<tr>
<td>3</td>
<td>Variable</td>
<td>Fixed</td>
<td>One</td>
</tr>
<tr>
<td>4</td>
<td>Variable</td>
<td>Variable</td>
<td>One</td>
</tr>
<tr>
<td>5</td>
<td>Variable</td>
<td>Variable</td>
<td>Multiple</td>
</tr>
</tbody>
</table>

Table 2.1: Combinations of the assumptions in the Thesis
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Variable</td>
<td>Mixed</td>
<td>Loop</td>
<td>Variable</td>
<td>Fixed</td>
<td>Variable</td>
<td>Network</td>
<td>Stations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Location</td>
<td>Free Flow</td>
<td>Tandem</td>
<td>Location</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Segmented</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2: Classification of models
empty travel considerations. The papers reviewed in this chapter are summarized based on their assumptions and solution procedure in Table 2.2.

In the following sections, papers are coded based on the initials of the authors followed by the last digit of the year of publication. The last two digits are used for papers published before 1990.

2.2.1 Loaded Vehicle Travels - Fixed Station Locations - Variable Network

A unidirectional flow network containing all arcs of the block layout is commonly known as the Conventional Configuration. In the models discussed in this section, it is assumed that the block layout and location of stations are fixed. The flow network is restricted to the edges on the boundary of the block layout. In most models stations are located on a set of new degree 2 nodes inserted on all edges, we refer to them as stations on junctions. In a few models they are on original degree 3-4 intersections, we refer to them as stations on intersections. The advantage of the first category of models is less congestion in real operations.

The objective of the model is to assign direction to the candidate edges in order to minimize the total loaded vehicle travel. The common elements of all these formulations are as follows:

\[ \text{Minimize} \quad \text{Loaded Vehicle Travel Distance} \]
\[ s.t. \]
\[ (a) \quad \text{Unidirectionality of Arcs} \]
\[ (b) \quad \text{Reachability of Nodes} \]
\[ (c) \quad \text{Strong Connectivity of the Flow Graph} \]
\[ (d) \quad \text{Binary Arc Variables} \]

Reachability of nodes states that there must exist at least one outgoing arc from, and at least one incoming arc to each node. A graph is strongly connected if it contains at
least one directed path from every node to every other node, [Ahuja, Magnanti, Orlin, 1993]. This is to avoid a trapped situation in which the vehicle could not leave a part of the connected flow network. To achieve strong connectivity, two constraints of at least one incoming and at least one outgoing arc are required for all connected sub-sets of nodes in which each node is adjacent to at least two other nodes. Alternatively, a dummy product with its production routing as a closed sequence of all stations may be introduced.

The flow path design was first formulated by Gaskins and Tanchoco [GT87]. In their Mixed Integer Programming model, the flow network and station locations are fixed, stations are located on the edges. The problem is to find the direction of arcs such that when the pre-determined shortest routes are taken, the total loaded vehicle travel distance is minimized. Since each P/D station has two incident arcs, there are four possibilities of leaving a P station, and entering a D station.

The decision variable $x_{ij}$ is 1, if the arc $ij$ is directed from nodes $i$ to node $j$, otherwise it is 0. The following objective function is then linearized using Lawler's substitution

$$Minimize \sum_m \sum_n f_{mn} \sum_{i=1}^4 (d_{mnp}X_{mq}X_{rn}) \quad q, r \in p \forall f \in F \quad (2.44)$$

$X_{mq}$ is the exit arc from node $m$, and $X_{rn}$ is entry arc to node $n$, when path $p$ is taken from $m$ to $n$. The parameter $d_{mnp}$ is the distance from station $m$ to station $n$ when the shortest path $p$ is taken.

Regarding each flow, four possible shortest routes, their length and number of arcs, are among inputs of the model. Nevertheless, identifying all these paths is a difficult task by itself. In addition, there are usually multiple stations per cell. The following additional constraints are required to ensure that the shortest routes are taken.

$$(n_p)X_{mq} + (n_p)X_{rn} - \sum_{mn \in p} X_{mn} \leq n_p$$

In the formulation for the above constraint, instead of $n_p$, they use $n_p - 2$, and then
exclude $X_{mq}$ and $X_{rn}$ from the third term. Shortest routes are identified outside the model under the assumption of a bi-directional network. After directing the network, it is possible to have a pair of P and D stations with none of their predetermined shortest routes being feasible, i.e., the feasible region is empty.

The constraints of unidirectionality of arcs, reachability of nodes, and strong connectivity of the flow network are also included.

All candidate arcs are finally assigned a direction, while some of them are never utilized. In Kaspi and Tanchoco [KT0] unlike [GT87], shortest routes are identified by the model. However, the number of the binary variables is increased by twice the number of edges times the number of flows. The number of constraints is also increased substantially. They develop an algorithm to satisfy reachability, and a more efficient branch and bound. Since no computational results are reported, it seems the model is restricted to very small problems.

Sinriech and Tanchoco [ST1] locate the P/D stations on the intersections on the block layout. An improved version of the branch and bound procedure developed in [KT0] is proposed. Computational results are limited to two examples. There is no indication of preprocessing or tightening the LP-relaxation.

Restricting station locations to the intersections on the block layout has two main advantages. (a) The search space is smaller. It is possible to develop more effective models and more efficient solution procedures. (b) Since the coverage of a degree 2 node guarantees the coverage of a node with greater degree, therefore, the optimal solution under the assumption of stations on intersections is at least as good as when stations are along the edges. However, its disadvantage is substantial, it results in higher blocking. Although analytical solutions under the assumption of locating the stations on the intersections are always as good as that of locating on junctions, simulation and real life experience show it does not necessarily remain the same in operations.

Chiang and Kouvelis [CK4] report that, unlike the previous models of [GT87],
[KT0], [ST1], [VW1], their MIP formulation can assign direction to arcs of layouts containing as many as 18 cells. The density of the flow graphs of the test problems are not reported. For problems larger than 18, the model fails to produce a solution in any period of time. For the purpose of strong connectivity, they introduce a dummy product with its production routing as a closed sequence of all stations. They developed different versions of simulated annealing and tabu search heuristics.

All the above models implement Branch and Bound in its conventional form of using an LP sub-problem.

2.2.2 Empty Vehicle Considerations - Fixed Station Locations - Variable Network

When an AGV carries out its material transport assignments, it travels empty from the D station to a P station. The total vehicle travel distance includes loaded and empty travel distances. Most of the studies on flow path design are restricted to loaded travel distance. Empty vehicle travel has a substantial impact the network design and location of stations.

To incorporate the impact of the empty vehicle travel, some dispatching rules must be implemented. In the majority of models, it is assumed that the vehicle remains at the delivery station until the dispatcher assign it to a new P station. In First Encountered First Served dispatching rule, Bartholdi and Platzman [1989], the vehicle is traveling and checks for a load in the next station encountered. In the model developed by Majety and Wang [MW5], the vehicle after completing its trip is sent to a central terminal. Venkataramanan and Wilson, [VW1] send the vehicle back to the P station it came from. The intention of the following papers is again to assign a direction to the arcs, while taking into account empty vehicle travel distances. [VW1] locate the stations on the intersections. They extend the model to include empty vehicle travel. In one assumption, empty vehicle travel is considered as a secondary objective function. Under the second assumption, both types of travel are of equal
importance. The empty vehicle returns from the D station back to the P station it came from. This is not the case in real operations. They adopt the branch and bound procedure developed by Little et al. [1963] for the traveling salesman problem. The Conventional Configuration differs from the TSP in its objective function, as well as the degree of nodes. Little et al. implement the general philosophy of Branch and Bound which is applicable on any sub-problem. In its adaptation to flow path design problem, [VW1], the sub-problem for the loaded vehicle travel is shortest route, while that of loaded/empty travel is shortest cycle.

[MW5] present a 0/1 nonlinear integer program to determine the flow path and terminal location. The nonlinear terms are later linearized. It takes one hour of CPU time to solve a 9 cell layout using a general purpose MIP solver on an IBM main frame.

Kouvelis, Gutierrez, and Chiang [KGC2] developed 5 heuristics and a simulated annealing implementation. Using the production routings and volume, and estimated intensity of the empty travel, a combined From-To chart for both types of travel is developed. Most of the heuristics are more or less of a greedy nature. They assign directions to arcs based on nonincreasing order of flows in the combined F-T chart, production volume of each part, the scalar product of production volume and the shortest route. The non-greedy heuristic starts by assigning directions based on the flows which create the least limitation on the shortest path of the remaining flows. They conclude that the simultaneous implementation of all 5 heuristics, and selecting the best, is more efficient than simulated annealing.

In this review we mainly concentrate on optimization models and conventional heuristics for designing material flow systems. We did not go into details of recent heuristics like Simulated Annealing, Generic Algorithm, Tabu Search.

Bozer and Srinivasan, [BS1], [BS2], conceptualized the tandem configuration. In their model, the layout as well as location of P/D stations are fixed, and the flow path is not restricted to the edges on the block layout. The flow path is partitioned into
a set of disjoint loops each served by a single vehicle. They developed a heuristic to generate promising feasible loops each covering a sub-set of P/D stations, and used a set partitioning formulation to find the optimal sub-set of the loops.

2.2.3 Loaded Vehicle Travel - Fixed Network - Variable P/D stations

The location of P/D stations is a major component of AGVs flow system design. They significantly influence the traffic intensity, the travel distance, and traffic control. Montreuil and Ratlif [MR88] developed a loaded vehicle travel optimization model to determine the location of P/D stations in a free flow environment. Multiple stations per cell were allowed, the station locations were not limited to the boundary of the cells. They decompose the problem into horizontal/vertical location determination. At each sub-problem, the layout excluding the node under consideration is merged into two nodes, one at left of (below) the node, the other at the right of (above) it.

Kiran and Tansel [KT89] model the problem of locating stations on a directed network. Two environments of stations on intersections and on edges are modeled. Pick-up and Delivery stations are combined.

Sinriech and Tanchoco [ST2], Tanchoco and Sinriech [TS2], and Sinriech and Tanchoco [ST3] developed two models to locate stations on a single loop.

Kim and Klein [KK6] consider the problem of finding P/D points on a given flow path. They formulate the problem as a QAP and suggest some heuristics. Their model is also extended to continuous station locations alongside the edges.

The objective function of all the above models is to minimize the total loaded vehicle travel.
2.2.4 Loaded Vehicle Travel - Variable Network - Variable P/D stations

Conventional Configuration

Goetz and Egbelu [GE0] integrate the determination of the direction of arcs and the location of stations. One P and one D station is allowed per cell. Their formulation is similar to that of [GT87] with multiplication of more than two variables in the objective function.

The definition of arc variables is the same as [GT87], [KT0]. The variable $Y_{ij}$ is 1 if the $j$th P location of cell $i$ is selected, and is 0 otherwise. The same is true regarding $W_{ij}$ and D stations. $n_i$ and $m_i$ are the number of P and D stations on cell $i$ respectively. $i_e$ is the set of cells receiving a major flow from cell $i$. Finally $d_{iklp}$ is the shortest path from $j$th P station of cell $i$ to $l$th D station of cell $k$ on path $p$. The objective function is

$$\text{Minimize} \quad \sum_{i=1}^{n} \sum_{j=1}^{n_i} \sum_{k \in i_e}^{m_k} \sum_{l=1}^{4} f_{ik} d_{iklp} Y_{ij} W_{kl} X_{mq} X_{rn} \quad (2.45)$$

Since they consider only one major flow per cell, there are $4n$ terms in the objective function, compared to $4n(n - 1)$ terms in [GT87]. Using Lawler's substitution, they linearize the multiplication of the set of the variables by a 9-subscript linear variable, and regulate the relationship between the linear and binary variables. We are unaware they use a 9-subscript instead of 8-subscript variable, and also locate it among the integers. Like [GT87] a set of constraints ensure that whenever the beginning and ending arcs of a path are selected, all intermediate arcs are also selected. They develop some rules to reduce the large number of the constraints. However, the large number of binary variables and no attempt to tighten the LP-relaxation leaves the model solvable for only very small problems. No computational results are reported.
Single Loop

A conventional configuration involves dispatching, vehicle routing, and traffic management problems. These may lead to excessive blocking as well as expensive software for operations. A simple configuration, like a single loop, does not have these difficulties. The simple dispatching rule of First Encountered First Served, Bartholdi and Platzman [1989], may be implemented in a single loop. Since there is only one route between every pair of nodes, vehicle routing is not required. Traffic management is much simpler, because there are no intersection on the flow path.

Afentakis [1989] proposed a loop layout for manufacturing systems. The loop configuration represents an attractive solution to flexible manufacturing systems layout. Its main advantages are simplicity and efficiency, low initial investment and expansion costs, product and processing flexibility, Afentakis [1989]. Bartholdi and Platzman [1989], suggested a single loop for Automated Guided Vehicle Systems. Kouvelis and Kim [1992], designed a unidirectional loop layout for automated manufacturing systems. Tanchoco and Sinriech [TS2], and Sinriech and Tanchoco [ST3] were the first researchers who developed a model for a loop configuration on a block layout. Since [ST3] is an improved version of [TS2], therefore we only refer to [ST3]. The objective of their model is to find the loop and the location of stations in order to minimize the total loaded vehicle travel distance. Their solution procedure contains five phases.

Phase-1, using a MIP model, a shortest loop is identified. They also propose a heuristic to find an initial loop. The heuristic starts with a cell having the maximum number adjacent cells. Its boundary forms a loop. If the loop does not cover at least one edge of every cell, a cell with at least one edge on the loop and adjacent to a cell with no edge on the loop, is selected. The covered edge of the cell is removed from the loop, and its uncovered edges are included.

In Phase-2, an enumeration procedure starts with the shortest loop or the feasible loop identified in phase-1. It enumerates all possible feasible and infeasible loops. If the layout contains \(|C|\) cells, dimensions of \(2^{\lvert C \rvert} - 1\) are required for the arrays in their
Phase-3, a set of 3 rules is applied on all enumerated loops to eliminate inferior loops. Rule-1; if loop-1 contains nodes of loop-2 in the same order, and loop-2 contains all cells of loop-1, then loop-1 dominates loop-2. Rule-2; if loop-1 contains all cells of loop-2, and loop-2 contains all nodes of loop-1 in the same order, and loop-2 is longer, then loop-1 dominates loop-2. Rule-3; if loop-1 contains all cells of loop-2, and loop-2 contains some nodes of loop-1 in the same order, and loop-2 is longer, then loop-1 dominates loop-2. For their 11-cell example, 17 loops are left in phase-3 out of 444 loops of phase-2.

During Phase-4, A mixed integer programming model is applied on each remaining loop of Phase-3. The model finds the direction of a fixed loop and location of P/D stations. The binary variable \( I_{(p_i,d_j)} \) is 0, if \( D_j \geq P_i \), and 1 otherwise. \( f_{(p_i,d_j)} \) is the flow from the P station of one cell to the D station of another cell. \( P_i, D_i \) are distances of location of stations from a reference point. \( U_i, L_i \) are the upper and lower bounds on locations of station of a cell. \( C \) is the length of the given loop. The distance between two stations is \( d_{ij} = (I_{(p_i,d_j)}C + D_j - P_i) \), which is coupled with

\[
D_j - P_i + I_{(p_i,d_j)}M \geq 0 \quad \forall i,j : f_{p_i,d_j} > 0
\]  

\[
L_i \leq P_i \leq U_i \quad i = 1, \ldots, M
\]  

\[
L_j \leq D_j \leq U_j \quad i = 1, \ldots, M
\]

If the boundaries of a cell is split into \( H \) parts along the loop. That is, the cell has \( H \) nonadjacent edges on the loop, then the above constraint is replaced by the following,

\[
P_i \leq U_i^{(h)} + M\delta_h^{P_i} \quad h = 1, \ldots, H
\]  

\[
P_i \geq L_i^{(h)} - M\delta_h^{P_i} \quad h = 1, \ldots, H
\]

\[
\sum_{h=1}^{H} \delta_h^{P_i} = H - 1
\]
\[ \delta^e_h = 0 \text{ or } 1 \quad \forall h, i, j \quad (2.52) \]

\[ I_{(p_i, d_j)} = 0 \text{ or } 1 \quad \forall i, j \quad (2.53) \]

A similar constraint is required for each delivery station.

The objective function is stated as

\[
\text{Minimize} \quad \sum_{i=1}^{M} \sum_{j=1}^{M} f_{(p_i, d_j)}(I_{(p_i, d_j)}C + D_j - P_i) \quad (2.54)
\]

The procedure is repeated for all remaining loops of phase-3. The model is applied on them one at a time. The optimal location of P/D stations of each remaining loop is identified. The solutions are compared, and the overall loaded vehicle travel optimal solution is identified. The model becomes more complex if more than one station is allowed per cell. This is the opposite of our model where increasing the number of stations decreases the complexity of the model.

Sinriech and Tanchoco [1992] conduct a simulation study to analyze the impact of empty vehicle flow on performance of a single loop. They show that empty vehicles, in contrast to their impact in conventional configuration, have negligible impact in a single loop configuration. A loop different from the optimal solution of their model is used. The conventional configuration selected for the purpose of comparison is also not the optimal one. We will show that the empty vehicle travel distance has a substantial impact on the optimal solution for a single loop material flow system.

### 2.2.5 A Cost Based Models

Chhajed, Lowe, and Montreuil [CLM2] developed a model to minimize the fixed cost of initiating a bi-directional network on a block layout, that excludes travel costs. The cost of the network is proportional to its length, stations are combined, one station per cell is allowed. Two environments of free flow, and flow on edges of the block layout are modeled. A Lagrangian relaxation and two heuristics are proposed.
Mixed Flow System

Goetschalls and Palliyil [GP4] developed an integrated model to add the flow network and P/D stations to a fixed layout. Their flow network is neither a specific configuration, nor restricted to be uni-directional. They allow multiple P, D, and combined stations per cell. The model accounts for empty vehicle travel. Two dispatching rules, one determined inside and the other outside the model are examined. The objective function contains fixed construction cost of aisles and stations, and the cost of loaded and empty vehicle trips. They examine primal/dual, arc/path, internal/external dispatching rule formulations. Construction heuristics based on criteria of flow, distance, and their scalar product are proposed. The solution time for the dispatching rule determined outside the model is substantially higher than that inside the model. Path formulation solution times are higher than arc formulation, dual simplex times are lower than that of primal. Solution times for a layout of size 10 is between 1000-34000 seconds. The 15 cell layout of Nugent et al [1968] takes 300 CPU hours without generating the optimal solution.

Segmented Flow Topology

Sinriech, Tanchoco, and Herer [STH6] proposed a segmented flow topology, where the network is divided into nonoverlapping and not necessarily connected zones. Each zone in turn is partitioned into one or more bi-directional segments. An individual vehicle operates in each segment, segments are not restricted to any specific pattern. Transfer buffers are located at the segment ends and serve as the interface devices between the segments. The flow system allows multiple stations per cell and stationary, trip based, and non-trip based material handling equipment. The objective is to minimize the sum of the fixed cost of initiating P/D stations and the travel costs of material handling equipment. However they do not follow an overall optimization procedure, since in some stages they use simulation to find a good feasible solution.
Their procedure consists of 5 steps.

Step 1. Like their earlier papers, stations are located on the intersections of the block layout. Dijkstra's algorithm is implemented to find the shortest route between all pairs of candidate P and D stations of each flow. In case of multiple shortest routes, which mainly happens when the origin and destination cells are adjacent, all multiple shortest routes are identified. This step identifies the set of candidate P and candidate D stations for each cell.

Step 2. A set of stations out of the candidate stations determined in step 1 are identified using a MIP model. The objective of the model is to minimize the fixed cost of initiating stations and loaded travel cost through the shortest paths identified in step 1. Since the travel term of their objective function is quadratic, the Lawler substitution is utilized to linearize them.

Step 3. Again Dijkstra's algorithm is implemented to find the shortest path of each flow using the selected station locations. An edge node incidence matrix for the selected nodes is formed. The rank of the matrix determines the number of the mutually exclusive zones. P/D stations and their incident edges are then assigned to specific zones. Based on loaded vehicle travel, and the time required for pick-up and delivery, the required number of vehicles at each zone are calculated.

Step 4. The concept of a single vehicle per segment proposed by Bozer and Srivinasan [1991] for a tandem configuration is implemented in this step. Each zone is partitioned into non overlapping segments, each containing a single vehicle. The required number of segments is equal to the number of carriers determined in step 3. Simulation is used to partitioned the zones into segments.

Step 5. Based on the number of mobile and stationary material handling equipment, the number of P/D stations, and travel distances, the total cost of the system is calculated.

The proposed configuration is compared with conventional as well as tandem configurations. Only one example is used to compare the three configurations. More
than 80 percent of the flows, both in terms of the number and volume of flow, are between adjacent cells. This situation, obviously motivates a segmented flow topology. Extensive experiments are required to compare tandem and segmented configurations.

The model is solved in 310 hours on a GOULD NP1 computer for an 11 cell example with a 15 percent density of F-T chart.

Sinriech, Tanchoco, and Herer [1996] propose segmenting a single loop where one AGV is operating in each segment in a bi-directional mode. The initial optimal loop is obtained using [ST3]. The segmentation is carried out using a heuristic, and the solution is improved through simulation. The primary goal is to minimize the segment to segment transfers, the secondary is to balance the workload among all segments.

In their simulation study, neither of segmented flow topology or tandem configuration dominates the other.
Chapter 3

The Shortest Loop for Non-trip Based Material Handling Systems

3.1 Introduction

The design of the material flow pattern is one of the main issues in planning a facility layout, Apple [1977], Tompkins et al., [1996]. Flow pattern designers usually avoid complicated networks. Four simple general flow patterns are identified as Straight line, Zigzag, U-shaped, and Loop, Apple [1977], Tompkins et al. [1996]. Afentakis [1989], developed a loop layout for manufacturing systems. The loop configuration represents an attractive solution to flexible manufacturing systems layout. Its main advantages are simplicity and efficiency, low initial investment and expansion costs, product and processing flexibility, Afentakis [1989].

One main objective function in designing the material flow pattern, especially in the case of non trip based material handling systems, is minimization of the total length of the flow path. Minimization of the length of edges on the flow network is an indicator of minimization of all costs directly proportional to the length of the network. These are initial fixed construction and installation costs, space occupied costs, maintenance and all similar one time or yearly costs. This objective is of crucial importance in designing material handling systems such as on ground conveyors like belt and roller, overhead conveyors like trolley, automated monorail, and power and
free. The main objective in designing these systems is the minimization of the length of the flow path.

In this chapter we develop a 0/1 Programming model to design a minimal length circular path along the edges of a block layout. The loop should cover at least one edge from each cell in the block layout. To the best of our knowledge, the only papers addressing the shortest loop design problem in a given block layout, are Tanchoco and Sinriech [1992], and Sinriech and Tanchoco [1993]. Our work differs from theirs in four main aspects.

1) The special structure of the block layout and its properties are exploited. The graph associated with the block layout by its exact dimensions is embedded in a plane. All its faces have right angle corners. Based on these properties, we develop a new formulation with fewer binary variables, and a different set of constraints.

2) A different sub-tour elimination approach is implemented. It is shown that the required number of sub-tour elimination constraints in our problem is much less than that of TSP. This is due to the special structure of the block layout, as well as a modified definition for sub-tour. Furthermore, the elimination constraints of the majority of the sub-tours are redundant.

3) The proposed sub-tour elimination constraints significantly tighten the feasible region of the LP-relaxation to the IP solutions. Consequently, the computation times are much lower than that of available models. For small to medium size problems, the optimal solution is found at the root node of the Branch and Bound.

4) We find the shortest loop directly.

3.2 The Problem

3.2.1 Formal Statement

Let $G_b(N, A)$ be the graph associated with the block layout, where $N$ is the set of intersections on the boundaries of cells, and $A$ is the set of nondirected edges. For
any adjacent pair of nodes in $N = \{ n_1, \ldots, n_{|N|} \}$, there exists an edge in $A = \{ n_i, n_j : i < j \text{ and } n_i, n_j \in N \}$. A distance or cost function $l : A \to \mathbb{R}^+$ assigns $l_{n_i,n_j}$ as the length of the edge connecting nodes $n_i$ and $n_j$. The length of each edge is equal to the rectilinear distance between its nodes. The graph with its weighted edges is embedded in the plane. Edges are either horizontal or vertical, therefore, the maximum degree of a node is 4. A feasible Loop is a cycle covering at least one edge of each face (manufacturing cell) \(^1\).

We will show that by a slight modification, a loop covering at least one edge of every cell is equivalent to a loop covering at least one node of them. The problem differs from the Hamiltonian Cycle problem (HC), in that HC must cover all nodes of each face, while it is enough for our loop to cover one node of each face. The Shortest Loop Problem (SLP) is to find the shortest feasible solution.

A heuristic for this problem is presented in Appendix A.

3.2.2 The Dual Representation

Let us define $G(C,A)$ as the Dual Graph (Adjacency Graph) of $G_b(N,A)$, in which each node $c \in C$ represents a cell of the block layout including the external face (cell). For each pair of adjacent cells $c$ and $k$ there is an edge $ck \in A$. There is a one to one correspondence, and equality of lengths between the edges of the primal and dual graphs. There is no restriction on the degree of the nodes in the dual graph. Faces are mostly triangular, while some are quadrilaterals. The primal graph of our 13-cell block layout, and its dual are shown in Figure 3.1.

There is a one to one correspondence between faces in the primal and nodes in the dual. Therefore, the "at least one edge of each cell" constraint is simply stated as "at least one edge of each node" in the dual graph. Any component of a solution in the dual graph contains at least 2 nodes. There is a one to one correspondence

\(^1\)The boundary of the layout, the external cell, should also have at least one edge on the loop. The physical interpretation of this assumption is the requirement for having a main Receiving and Shipping point on the boundary of the manufacturing plant.
Figure 3.1: Primal and Dual graphs of the 13-cell prototype example.
between nodes in the primal and faces in the dual. The “at most degree 2 node” constraint, which is a necessary condition for having a loop in the block layout, is stated as “at most 2 edges of each face” in the dual. Each face of the dual could have either no edge or two edges in a solution. By a solution we mean a single loop or a set of disjoint loops or a sub-graph covering at least one edge of each cell. By a feasible solution we mean a single loop covering at least one edge of every cell.

Given a not necessarily connected flow path covering at least one edge of every cell in the block layout, its corresponding dual representation is shown in Figure 3.2(a). Given a degree 2 flow path in the block layout, the corresponding dual representation is shown in Figure 3.2(b). Both these instances are solutions, Figure 3.2(c) is the dual representation of a feasible solution. Figure 3.2(d) shows that a feasible loop in the block layout, is not necessarily a connected sub-graph of the dual.

### 3.3 Transformation to the Generalized Traveling Salesman Problem

#### 3.3.1 The Generalized Traveling Salesman Problem

The Generalized Traveling Salesman Problem (GTSP) is a well-known extension of the Traveling Salesman Problem (TSP). The GTSP can be formally defined as follows. Let $G_b(N, \bar{A})$ be a graph where $N = \{n_1, \ldots, n_{|N|}\}$ is the vertex set, and $\bar{A} = \{n_i, n_j : i \neq j$ and $n_i, n_j \in N\}$ is the set of arcs. A distance or cost function, $l : \bar{A} \rightarrow \mathbb{R}^+$, assigns $l_{n_i, n_j}$ as the distance or cost of travel from node $n_i$ to node $n_j$. If the length or cost matrix is symmetric, then the directions are irrelevant, and the graph is defined as $G_b(N, A)$, where $A$ is the set of non-directed edges. The set $N$ is the union of $|C|$ clusters, e.g., $N = \{N_{c_1} \cup \ldots \cup N_{c_{|C|}}\}$. The problem consists of determining the shortest Hamiltonian Cycle passing through each cluster at least once. In some versions of the GTSP, the condition “at least once” is replaced by “exactly once”.

The GTSP was introduced by Henry-Labordere [1969], Saksena [1970], and Sri-
Figure 3.2: Dual graph representation of loop constraints
vastava et al. [1969]. The applications were in relation to record balancing problems arising in computer design and routing of clients through agencies providing various services. Laporte and Nubert [1983] proposed a constraint relaxation algorithm for the symmetric version of the problem. A Lagrangian relaxation algorithm valid for symmetric version is presented by Noon and Bean [1991], while Fischetti, Salazar and Toth [1994] developed a branch and cut algorithm for symmetric instances. An interesting construction is that of Noon and Bean [1993]. These authors show that the GTSP with disjoint clusters defined on a directed graph can readily be transformed into an equivalent TSP involving the same number of vertices. This finding does not imply, however, that the asymmetric GTSPs can be solved as efficiently as symmetric TSPs. The transformation induces a fair amount of degeneracy. Laporte, Asef-Vaziri, and Sriskandarajah [1996] showed that a wide variety of combinatorial optimization problems can be modeled as GTSPs. Location-routing problems, post box collection, stochastic vehicle routing, and arc routing are among these problems.

3.3.2 The Transformation

Consider the block layout of a production plant partitioned into $c_1, c_2, \ldots, c_{|C|}$ cells. Let us first assume that the objective of the problem is to find the shortest loop with at least one node of each cell in the block layout.

By defining the vertex set $N_c$ as the nodes on the boundary of cell $c$, the problem is easily transformed into the GTSP. We show that by a slight modification, finding a loop covering at least one edge of each cell is equivalent to the loop covering at least one node of the cells.

**Theorem 1** The problem of finding a loop with at least one edge of each cell is a Generalized Traveling Salesman Problem.

**Proof:** Insert a new node on each edge of the primal graph associated with the block layout. Redefine the nodes of each cluster as the set of the new degree 2 nodes on
each cell. The problem is an instance of the GTSP. We show that insertion of these new degree 2 nodes are only required for pairs of adjacent degree 4 nodes.

**Definition 1** A degree $k$ node, $k = 2, 3, 4$ lies at the intersection of $k$ cells. The cells adjacent to a node are its defining cells. The intersection of each pair of defining cells is a defining edge.

Suppose all nodes of the block layout are of degree 3. When a loop covers a node, a pair of its defining edges are covered. Each edge belongs to a unique pair of defining cells. Two defining edges of a node belong to all of its three defining cells. Whenever a node is covered, at least one edge of all its defining cells are covered too.

The proof is easily extended to degree 2 nodes. Whenever a degree 2 node is covered, both of its defining edges are covered. Since a degree 2 node has two defining cells, both defining cells have at least one edge on the loop. However, degree 2 nodes need not be included in the clusters at all. Whenever a degree 2 node is covered, both of its adjacent non-degree 2 nodes are covered too.

The case of degree 4 nodes is not that clear. For example, the loop covering nodes $n_1, n_2, n_3, n_4$ in Figure 3.3(a), contains a vertex of cell $c_4$, but not an edge of it. Some adjustments are required for such vertices.

Degree 4 nodes not adjacent to any other degree 4 node are simply removed from
all the clusters $N_c$ they belong to. This transformation guarantees that whenever a node of a cell is covered by a loop, an edge of it is also covered. Furthermore, it does not restrict degree 4 nodes from being on the loop.

Consider in turn those edges $n_i, n_j$, where both nodes $n_i$ and $n_j$ are of degree 4. A new node $n_o$ is inserted somewhere on the corresponding edge. Both $n_i$ and $n_j$ are removed from all their clusters, and node $n_o$ is added to the clusters that they both belong to. In Figure 3.3, a new vertex would have to be introduced between $n_1$ and $n_2$, but not between $n_2$ and $n_3$. Node $n_1$ is removed from clusters $c_1, c_2, c_4, c_5$, node $n_2$ is removed from clusters $c_2, c_3, c_5, c_6$ and node $n_o$ is added to clusters $c_2, c_5$. Under this transformation, the shortest loop with at least one edge on each cell can be solved as a GTSP, where it is required to go through each of the modified clusters $N_1, \ldots, N_{|C|}$ at least once.

3.4 Complexity

Feasibility Is there a feasible solution to The Shortest Loop Problem?

Optimality. Suppose construction cost of the loop is an adequate measure of effectiveness, and is a linear function of the length of the loop. Optimize $Z = \sum_{mn \in A} l_{mn} Y_{mn}$, under the constraints stated for SLP. Where $Y_{mn}$ is the binary decision variable associated with the nondirected edge $mn$, $n > m$. The variable is equal to 1 if the edge is on the loop, and 0 otherwise.

The SLP belongs to $NP$. Feasibility of any given loop is verified in polynomial time.

Property 1 Given $G(C, A)$ as the adjacency graph of the block layout. Suppose $s$ and $C \setminus s$ are the set of cells inside and outside the loop respectively. The problem of finding a loop with at least one edge on each cell is to find a sub-set of nodes $s$ such that

\[^2\text{In most of the remainder of this thesis, instead of the notation of } n_i \text{ for nodes and } c_i \text{ for cells, single indices like } n \text{ and } c \text{ are used.}\]
i) $s$ is connected.

ii) $\forall c \in s, \exists k \in G \setminus s \Rightarrow ck \in A$.

iii) $C \setminus s$ is connected.

iv) $\forall k \in C \setminus s, \exists c \in s \Rightarrow ck \in A$

In words, a loop partitions the set of cells into two sub-sets. It also touches one edge of every cell. Therefore, every cell inside the loop must be adjacent to a cell outside the loop, and vice versa.

If any of the above conditions is violated, then there is a set of disjoint loops, not a single loop.

### 3.4.1 Existence of a Feasible Loop

The problem SLP may have an empty feasible region. Figure 3.4 shows a 5-cell block layout with no feasible loop. Indeed to visit cell $c_3$ and $c_5$, edges $a, b, c, d$ must be covered, which by itself forms a loop, and leaves no place for an edge of cell $c_1$ to be covered. The problem has no feasible solution.

By a simple transformation, any non-hamiltonian planar graph with maximum degree of 3 may be transformed into a Block layout with no feasible loop. This is done by replacing each node with a rectangle, and each edge by a rectilinear line. This is usually possible by a two piece rectilinear line, and always possible by a rectilinear line with more than two pieces. Adjacent edges of the graph are connected to distinct nodes of the rectangle. The resulting block layout does not contain a feasible loop.

### 3.5 Lower Bounds

#### 3.5.1 The Number of Edges

**Property 2** The lower bound for the number of edges in a feasible loop is $\max\{(|C| - |N_4| - 1), ([|C|/2])\}$, where $N_4$ is the set of degree 4 nodes.
Figure 3.4: An example of a block layout with no feasible loop

Proof: A feasible solution contains at least one edge of each cell. Each edge belongs to a unique pair of cells. (It is not necessary for a pair of adjacent cells to be defined by only one edge.) Therefore, to cover |C| cells, at least \(||C|/2\)| edges must belong to the loop. The only case in which each new edge could cover a pair of new cells is when there is no degree 3 node on the loop. Two examples of the class of the block layouts, where \(||C|/2\)| is the lower bound for the number of edges on a feasible loop are shown in Figure 3.5. There is a tighter lower bound when \(|N_4| < ||C|/2\) - 1.

A degree \(k\) vertex has \(k = 3\) or 4 defining cells. Given an edge connected to a degree \(k\) node, by deleting \((k - 2)\) already covered defining cells, the intersection of the remaining \((4 - k)\) already covered and \((k - 2)\) not covered defining cells is an edge. Moving along this edge will cover \((k - 2)\) new cells. If \(N_k^k\) is the set of degree \(k\) nodes on the loop, the following relationship holds

\[
\sum_{k=3}^{4} (k - 2)|N_k^k| \geq |C| - 1
\]
Figure 3.5: Two examples of the class of the block layouts with $\lceil |C|/2 \rceil$ as the lower bound for the number of edges on a feasible loop.
Figure 3.6: Two examples of the class of the block layouts with $|C| - |N_4| - 1$ as the lower bound for the number of edges on a feasible loop.

$$|N_3^L| + 2|N_4^L| \geq |C| - 1$$

It is $|C| - 1$ on the right hand side because the starting node is assumed to be degree 3. Otherwise all nodes are degree 4 and $\lceil |C|/2 \rceil$ is a tighter lower bound. The first edge initiating from a degree 3 node covers two new cells. By taking one $|N_4^L|$ to the right hand side,

$$|N_3^L| + |N_4^L| \geq |C| - |N_4^L| - 1$$

The left hand side is the number of edges on the loop. If all degree 4 nodes are covered by the loop, then the lower bound for the number of edges is $|C| - |N_4| - 1$.

### 3.5.2 Length

A sub-graph of $G_b(N, A)$ containing the first $\max\{(|C| - |N_4| - 1), (\lceil |C|/2 \rceil)\}$ shortest edges is a lower bound for the length of the loop. The sub-graph is not necessarily connected.
3.5.3 Number of Cells

The lower bound for the minimum number of cells inside a feasible loop is expressed as

\[ \text{Min} \quad z = \sum_{c \in C} x_c \]

s.t.

\[ \sum_{c \in C} (|\mathcal{Y}_c| x_c) \geq \max\{(|C| - |N_d| - 1), (\lceil |C|/2 \rceil)\} + 2(z - 1) \]

where \( \mathcal{Y}_c \) is the set of nondirected edges on cell \( c \). The binary variable \( x_c \) is 1 if cell \( c \) is among the selected sub-set of cells, and 0 otherwise. If the term \( 2z \) is removed from the constraint, then the problem is a knapsack problem. The coefficients of all variables are equal in the objective function. Therefore a heuristic to solve the above problem is a simple modification of Bang for buck. Sort the cells in non-increasing order of the number of their edges. By starting from the cell with maximum number of edges, following cells are added until the above inequality is satisfied.

3.6 Loop Flow Pattern

3.6.1 Formulation

Formulation-1

A binary variable \( X_{mn} \) is defined for the directed arc \( mn \). The variable is equal to 1 if the directed arc is on the loop, and 0 otherwise. The formulation is stated as follow. Each cell has at least one arc in the final loop.

\[ \sum_{m, n \in \mathcal{N}_c} X_{mn} \geq 1 \quad \forall c \in C \quad (3.1) \]

Arcs are unidirectional

\[ X_{mn} + X_{nm} \leq 1 \quad \forall mn \in A \quad (3.2) \]

At most one incoming arc to each node.

\[ \sum_{mn \in \bar{A}} X_{mn} \leq 1 \quad \forall n \in N \quad (3.3) \]
\( \overline{A} \) is the set of directed arcs. The number of incoming arcs at each node is equal to the number of outgoing arcs.

\[
\sum_{mn \in \overline{A}} X_{mn} = \sum_{nm \in \overline{A}} X_{nm} \quad \forall n \in N
\]  

\[
X_{mn} \in \{0, 1\} \quad \forall mn \in \overline{A}
\]

This formulation is adopted from Sinriech and Tanchoco [1993].

**Formulation-2**

Since the distance matrix is symmetric, therefore, instead of two binary variables, one for each direction of the arcs, only one binary variable is defined for the corresponding nondirected edge. The binary variable \( Y_{mn} \) where \( n > m \) is equal to 1, if the nondirected edge \( mn \) is on the loop, and 0 otherwise. However, an additional set of binary variables corresponding to the nodes of the layout are required. The binary variable \( u_n \) is equal to 1, if node \( n \) is on the loop, and is 0 otherwise.

Each cell has at least one edge on the loop.

\[
\sum_{mn \in Y_c} Y_{mn} \geq 1 \quad \forall c \in C
\]  

Nodes are either of degree 0 or 2.

\[
\sum_{m < n} Y_{mn} + \sum_{k > n} Y_{nk} = 2u_n \quad \forall n \in N
\]

This formulation is adopted from the Generalized Traveling Salesman Problem formulation, Laporte and Nubert [1983].

**Formulation-3**

Although the cells of the block layout need not be convex, they are all right angle polygons. The degree of nodes is therefore restricted to 4. This property, coupled with the symmetry of the distance matrix, is utilized to reduce the number of binary variables and to formulate the degree 2 constraint.
Like Formulation 2 only one binary variable is used for each nondirected edge, but no binary variables are required for nodes.

Equation 3.6 remains unchanged.

At most 2 edges incident to a node are on the loop.

\[ \sum_{m<n} Y_{mn} + \sum_{n<k} Y_{nk} \leq 2 \quad \forall n \in N \]  

(3.8)

The number of edges on the loop at each node is not one. In other words, at each degree \( k \) node, the sum of the decision variables corresponding to each set of \( k - 1 \) edges is greater than or equal to the remaining one.

\[ \sum_{i<m} Y_{im} + \sum_{i>n} Y_{ni} \geq Y_{mn} \quad \forall mn \in A \]  

(3.9)

\[ \sum_{i<n} Y_{in} + \sum_{i>m} Y_{mi} \geq Y_{mn} \quad \forall mn \in A \]

\[ Y_{mn} \in \{0, 1\} \quad \forall n > m \]

The number of these constraints in a block layout with \(|N|\) nodes, is in the range of \(3|N|\) to \(4|N|\). If each cell on the average contains 5 edges, then \(|N|\) is almost \(1.5|C|\).

Therefore, the number of these constraints in our problem is less than \(7|C|\), compared to \(|C|^2\) in TSP.

### 3.6.2 Degree 4 cycles

The loop may be allowed to have degree 4 nodes. Figure 3.7 shows that a degree 4 node could split into two degree 2 nodes, and still retain the loop properties. To allow such a configuration, constraint 3.8 is removed on degree 4 nodes, and the following is added to prevent having 3 edges of these nodes on the flow pattern.

\[ \sum_{i<m} (1 - Y_{im}) + \sum_{i>n} (1 - Y_{ni}) \geq (1 - Y_{mn}) \quad \forall mn \in A : \forall m \in N_4 \]  

(3.10)

\[ \sum_{i<n} (1 - Y_{in}) + \sum_{i>m} (1 - Y_{mi}) \geq (1 - Y_{mn}) \quad \forall mn \in A : \forall n \in N_4 \]
3.6.3 A Covering Inequality

Any feasible loop contains at least \( \max\{(|C| - |N_4| - 1), (\lfloor |C|/2 \rfloor)\} \) edges. If \(|N_4| < |C|/2 - 1\), the following constraint is added to the problem to cut a portion of the feasible region of the LP-relaxation.

\[
\sum_{mn \in A} Y_{mn} \geq |C| - |N_4| - 1
\]  

(3.11)

The constraint is redundant when \(|C|/2 \leq |N_4|\). This is a direct consequence of constraint 3.6 which states each cell must have at least one edge on the loop.

\[
\sum_{mn \in Y_c} Y_{mn} \geq 1 \quad \forall c \in C
\]

The sum of the above constraint over all cells results in

\[
\sum_{c \in C} \sum_{mn \in Y_c} Y_{mn} \geq |C|
\]

Since each edge belongs to two cells, the left hand side of the above constraint is twice the number of edges in the block layout. Therefore,

\[
\sum_{mn \in A} Y_{mn} \geq |C|/2
\]  

(3.12)

The "at least one edge constraint", 3.6, is always included in the model. By comparing 3.11 with (3.12) redundancy of the first constraint is proved for \(|N_4| \geq |C|/2 - 1\).
3.7 Sub-Tours

3.7.1 Formation of Sub-Tours

The constraints of formulations 1-3 ensure formation of a degree 2 configuration, but do not guarantee a single loop. A set of disjoint loops may appear. Details of patching the potential sub-tours is discussed in this section. In the Traveling Salesman Problem, a sub-tour is defined as a loop covering a sub-set of cities. The definition of sub-tour in our problem differs slightly from its general form in TSP.

Definition 2 Any connected sub-set of adjacent cells is a potential sub-tour. The sub-tour is on the boundary of the composite shape.

A sub-tour becomes a feasible loop if its composite shape boundary covers at least one edge of every cell. Two sub-tours are shown in Figure 3.8(a). A feasible loop is shown in Figure 3.8(b).

The Number of Sub-Tours

Property 3 The upper bound for the number of sub-tours in a symmetric Traveling Salesman Problem is $2^{|C|} - |C|^2/2 - |C|/2 - 1$.

Proof: Since each city is either in a sub-tour or not, for $0 \leq |s| \leq |C|$ there are $2^{|C|}$ combinations. However, only combinations of up to $|C|/2$ cities require sub-tour elimination constraints. Any larger sub-tour $s$ does not require its own constraint because the remaining sub-tour on $C \setminus s$ is already included in the model. Zero city and one cities may not form a sub-tour in a TSP. Furthermore, in a symmetric traveling salesman, when the binary variable $X_{mn}$ is only defined for $n > m$, two city sub-tours are also not possible. Therefore, the total number of sub-tours excluding zero, one, and two city sub-tours is equal to

$$
\sum_{k=3}^{\lfloor |C|/2 \rfloor} \binom{|C|}{k} = \sum_{k=3}^{\lfloor |C|/2 \rfloor} |C|!/(|C| - k)!k! = 2^{|C| - 1} - |C|(|C| - 1)/2 - |C| - 1
$$
In our problem, since one and two cell sub-tours are possible, the upper bound for the number of sub-tours is $2|C| - 1 - 1$.

**Property 4** The number of potential sub-tours in a block layout is less than $2|C| - 1 - 1$.

The adjacency graph of the block layout reflects an important property of SLP. The adjacency graph of the TSP and the GTSP is theoretically complete, i.e. each cell is adjacent to $|C| - 1$ cells. Therefore $2|C| - 1 - 1$ is the exact number of sub-tours.

The adjacency graph of our problem is planar, and far from complete. For a planar graph there does not exist more than $3|C| - 6$ pairwise adjacencies, Foulds [1992]. Therefore, even in the case of maximum adjacencies, each cell on average is adjacent to less than 6 other cells.

**Number of 1-4 Sub-tours**

**Theorem 2** The number of potential sub-tours containing 1-4 cells is linear in the number of cells.

Proof: The number of 1-cell sub-tours in our problem is $|C|$.

The number of edges, is the upper bound for the number of 2-cell sub-tours. Note that degree 2 nodes have no structural meaning in our formulation. Whenever a piece-
wise line has only two adjacent cells, it is considered as one edge. The intermediate node is not included in the corresponding graph.

The above upper bound is tight. The only class of problems in which the number of edges is not equal to the number of two cell sub-tours is where two cells have more than one edge in common. The exact number of 2-cell sub-tours is the number of edges of the dual graph, where repetition of edges is not allowed. That is, each pair of cells is connected by at most one edge. The total number of 2-cell sub-tours for the prototype example of Figure 3.1 is 34, which is equal to the number of its edges. Given the number of cells and nodes of the primal graph associated with the prototype example, the number of edges is calculated using the Euler formula, Bondy and Murty [1976].

\[ |A| = |C| + |N| - 2 \]

The number of 3-cell sub-tours in a block layout is bounded above by

\[ \sum_{c \in C} \left( \frac{y_c}{2} \right) \]

where \( y_c \) is the number of edges on cell \( c \). However the more degree 3 nodes a graph has, the fewer sets of 3 connected cells it has. Actually this is fairly easy to see. Each cell adjacent to a degree 3 node, based on the above formula, appears 3 times in a unique sub-tour. Therefore, the following is a better upper bound.

\[ \sum_{c \in C} \left( \frac{y_c}{2} \right) - 2N_3 \]  \hspace{1cm} (3.13)

In the prototype example, there are 22 nodes, two are degree 4. There are three cells with 3 edges, three cells with 4 edges, five cells with 5 edges, two cells with 6 edges, and one, the external cell, with 10 edges. The exact number of 3-cell sub-tours is 112.

Instead of counting the number of edges on each cell, an average number may be applied. We are unaware of any study on the relationship between the number of edges and cells on practical layouts or layouts presented in the literature. However,
analysis of the majority of the layouts in the well known books on facilities planning including Francis and White [1974], Apple [1978], Tompkins and White [1984], Francis, McGinnis, and White [1992], Tompkins et al.[1996], and also layouts in literature, show that this ratio is around 2.5, i.e. 5 edges per cell. The number of degree 4 nodes is around 10 percent. Based on this estimate of 5 edges per cell, the number of 2-cell sub-tours is $2.5|C|$. Using the Eular formula and 90 percent degree 3 nodes, the estimates of $|N| = 1.5|C|$ and $|N_3| = 1.35|C|$ are obtained. It results in $9|C|$ as the estimated number of 3-cell sub-tours. The estimated number of potential 3-cell sub-tours for the prototype example is 100 compared to its actual number of 112.

An upper bound for the number of 4-cell sub-tours is

$$
\sum_{c \in C} \left( \frac{|Y_c|}{3} \right) + \sum_{c \in A} (|Y_c| - 1)(|Y_k| - 1) \tag{3.14}
$$

There are two possibilities for a 4-cell sub-tour to appear. (a) The combination of a cell with 3 of its adjacent cells; that is the first term in the above expression. (b) The combination of a pair of adjacent cells with two other cells each adjacent to one of them; that is the second term. The average number of potential 4-cell sub-tours based on the average of 5 edges on each cell is $50|C|$.

The accuracy of these estimates is sensitive to the variance of the number of edges per cell.

### 3.7.2 Generation of Potential Sub-tours

To identify potential sub-tours, an "Oriented-Not-Duplicated" version of the dual graph is traversed, and all its connected sub-graphs are enumerated. This is the dual graph where only one edge exists for each pair of adjacent cells $c$ and $k$. The edge is oriented from $c$ to $k$, where $k > c$. This is to avoid duplication.

The potential sub-tour generation process, identifies each cell as a potential sub-tour. Then it enumerates all 2-cell sub-tours initiated from each cell. For example, 2-cell sub-tours initiated from cell $c_4$, are all edges of the star with $c_4$ as its center.
Figure 3.9: Representation of a generation of sub-tours
A Loop Material Flow System Design for Manufacturing Plants

Table 3.1: The adjacency matrix of our 13-cell example

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
<th>C10</th>
<th>C11</th>
<th>C12</th>
<th>C13</th>
<th>C14</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>C2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>C6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>C7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>C8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>C9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>C10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>C11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>C12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>C13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>10</td>
<td>34</td>
</tr>
</tbody>
</table>

The adjacency matrix is utilized to identify all stars, i.e., all possible combinations of 1,2,... adjacent cells.

**Definition 3** Given the oriented-not-duplicated dual graph $G(C, E)$, where $E$ is the set of single oriented edges $c_k$, $c < k$ connecting node $c$ to its adjacent nodes $k$. The upper diagonal adjacency matrix $A = (a_{ck})_{|C| \times |C|}$ of the graph is the matrix with $a_{ck} = 1$ if $c_k \in E$, and $a_{ck} = 0$ otherwise.

The adjacency matrix of the prototype example is shown in Table 3.1. The number of 2-cell sub-tours is equal to the sum of the elements of the adjacency matrix.

**Definition 4** Each row of the Augmented Adjacency Matrix $AAM_1^{|s|}$ corresponds to a unique sub-tour $s$ of size $|s|$. The elements of each row $AAM_1^{|s|}$ of this matrix are the union of the adjacency rows and columns of all cells $c \in s$ excluding themselves. This row identifies the children of sub-tour $s$, which in turn are the rows of the next augmented adjacency matrix of size $|s| + 1$. 
For instance, the row $AAM_{c_5c_6c_9}^3$ has 1 in columns 1,4,7,8,10,13, and 0 in other columns, i.e., it is adjacent to $c \in \{c_1, c_4, c_7, c_8, c_{10}, c_{13}\}$. The process is initialized at $|s|=1$ by copying the adjacency matrix into the augmented adjacency matrix. The pseudo-code of the generation process of the potential sub-tours is shown in Appendix A.

### 3.7.3 Size of the Sub-Tours

**Property 5** The process of generating sub-tours does not necessarily continue up to its largest possible size of $|C|/2$. The size of a sub-tour is controlled by two thresholds. It may become an impossible sub-tour or a feasible loop.

The generation process initiated from each cell reaches its end in the following cases.

**Case 1**) $\exists c \in s \rightarrow \forall k \in C \setminus s \Rightarrow ck \notin A$.

This is when a cell and all its adjacent cells are in the sub-tour. The formation of such a sub-tour in the presence of the “at least one edge” constraint is impossible. Any subsequent sub-tour of this generation inherits this attribute, its formation is impossible. The process reaches its end.

**Case 2**) $\forall k \in C \setminus s \rightarrow \exists c \in s \Rightarrow ck \in A$.

In this case, all cells not in the sub-tour are adjacent to at least one cell in the sub-tour. Therefore it is a feasible loop. As soon as a sub-tour becomes a feasible loop, its process terminates.

A child of a feasible loop could stay only in 2 states: 1) All of its cells still have at least one edge on the boundary of their composite shape i.e., it remains a feasible loop. 2) At least one of its cells does not have an edge on the boundary, i.e., it becomes an impossible sub-tour.

**Property 6** A sub-tour of size $|s| < |C|$ could form a feasible loop.

Indeed in our problem, unlike the TSP, there is no tour containing all cells, and only one, whether possible or not, containing all internal cells. Indeed, a considerable
Figure 3.10: A feasible loop containing 3 cells in the 11-cell example

number of $|s| < |C|$ adjacent cells are feasible loops in our prototype examples. Figure 3.10(a) shows a feasible loop containing 3 cells in the 11-cell example. Obviously, no feasible loops should be eliminated.

To limit the extent of the search process, it is useful to find the smallest feasible loop as well as the smallest impossible sub-tour.

The smallest number of cells in a feasible loop was identified earlier using a knapsack formulation.

The smallest impossible sub-tour corresponds to the cell with the smallest number of adjacent cells and all its adjacent cells.

The pseudo-code of the algorithm to identify the termination point of each generation is in Appendix B.

3.7.4 Primitive Sub-tours

Any sub-tour partitions the Adjacency Graph into 3 sub-graphs.

$s = \{ \text{set of cells in the sub-tour} \}$

$s = \{ \text{set of cells adjacent to } s \text{ in } G_x \}$

$\bar{s} = C \setminus (s + s)$

$G_x(C, A_x)$ is an extended version of $G$ where two cells with only one common node in the block layout are also assumed adjacent. Note that $s$ is identified based on
adjacencies in $G_x$, while the identification of $\bar{s}$ is based on adjacencies in $G$. The sub-set $\bar{s}$ is the full complement of $s$. Any dominating sub-set $\bar{s} \subseteq \bar{s}$ is a potential sub-tour.

Given a graph $G(C, A)$, $C' \subseteq C$ is a dominating sub-set if every cell $c \in C \setminus C'$ is joined to at least one member of $C'$, Garey and Johnson [1970].

This differs from the TSP where each sub-tour has only one complement, and the complement is connected.

The relationship between sizes of a sub-tour and its full complement in our problem differs from that of the TSP. Indeed, $|s| + |\bar{s}|$ is not only less than $|C|$ but it is not even constant. The static criteria of $|s| \leq \lfloor |C|/2 \rfloor$ for a sub-tour to be Primitive, i.e., its elimination constraint to be included in the model, is replaced by the dynamic criteria of $|s| \leq |\bar{s}|$.

Furthermore, the sub-graph induced by $\bar{s}$ is not necessarily connected. Given a graph $G(V, E)$, let $V'$ be a non-empty sub-set of $V$. The sub-graph of $G$ whose vertex set is $V'$ and whose edge set is the set of edges of $G$ that have both ends in $V'$ is called the sub-graph induced by $V'$ and is denoted by $G[V']$, Bondy and Murty [1976].

**Property 7** A sub-tour $s$ is primitive, if its size is smaller than the size of the smallest component $r$ of its complement. The case of equality is decided based on the smallest identification number of the cells in $s$ and $r$.

This is explained using the following simple implications

1) $IF \ X \Rightarrow Y \land Z \ \THEN \ (\neg Y) \lor (\neg Z) \Rightarrow \neg X$

2) $IF \ X \Rightarrow Y \lor Z \ \THEN \ (\neg Y) \land (\neg Z) \Rightarrow \neg X$

The search for primitive sub-tours proceeds in non-decreasing order of the number of cells in the sub-tours. Within each specific size of sub-tours, it starts with the sub-tour initiated from the cell with smaller identification code.

The first rule states, if the full complement of a sub-tour $s$ is a disjunct set of sub-tours, elimination of any of these later sub-tours leaves no place for the sub-tour.
s to appear. Consider sub-tour \( s = \{A, D\} \) in Figure 3.11(a). Due to “at most degree 2” and “at least one edge” constraints, formation of this sub-tour certainly implies formation of three sub-tours of F, K, M. If the elimination constraint for any of these later sub-tours is included in the model, there is no need for the elimination constraint of sub-tour \( s \).

The second rule states, if the full complement of a sub-tour \( s \) is a conjunct set of sub-tours, and if all these sub-tours are already eliminated, then sub-tour \( s \) never appears. Consider the sub-tour \( s = \{C, H, J\} \) in Figure 3.11(b). Formation of \( s \) leads to formation of A or L or AL sub-tours. If 1-cell and 2-cell sub-tours are already eliminated, sub-tour \( s \) will never appear.

To unify the above implications, consider sub-tour \( s = \{A, B, L\} \) in Figure 3.11(c). Because of the “at least degree 2” constraint, the external edges incident to the boundary of the sub-tour could not belong to any sub-tour. These are a sub-set of edges on cells C,D,G,K, as well as cell H. On the other hand, “at least one edge” constraint forces cells F,J,M to have at least one edge on the flow pattern. This inevitably causes formation of sub-tour F and one of J, M, or JM. As soon as a sub-tour elimination constraint for either F or (J and M and JM) is/are included, there is no place for sub-tour \( s \) to appear. The reverse is not true, that is, elimination of \( s \) does not ensure elimination of F, J, M, and JM sub-tours.

Given a sub-tour \( s \), the sub-graph induced by \( s \) is not necessarily connected. Suppose \( r \) is the smallest component of \( s \), whenever \( |s| < |r| \) the sub-tour is primitive. If \( |s| = |r| \), the smallest identification number (letter) of the cells in \( s \) and \( r \) are compared. Suppose these identification codes are \( c \) and \( k \) respectively. If \( k < c \), then the sub-tour is not primitive.

Two properties of “\(|s| + |\bar{s}| < |C|\)” and “\( s \) not necessarily connected” are additional justifications for having a smaller number of primitive sub-tours than that of TSP.

The computational time required to generate a sub-tour is less than checking for its primitivity. Each sub-tour containing the external cell has a complement not including
Figure 3.11: Illustration of main characteristics of sub-tours
this cell. We have not included the external cell in the “oriented-not-duplicated” dual
graph. Instead of generating sub-tours containing the external cell and checking them
for primitivity, their complements are assumed all primitive. Other sub-tours, which
are adjacent to the external cell, are checked for primitivity.

Recall the discussion on the size of sub-tours in Section 3.7.3. Given a cell c, and
set B as the set of cells on the boundary of the block layout, as long as a sub-tour
does not contain a cell in B, its generation could grow up to the size of |C| − |B| − 1.
Otherwise, it grows up to the size of ⌊|C|/2⌋.

Whenever a sub-tour is primitive, its elimination constraint is added to the model.
The procedure starts with 1-cell sub-tours, after evaluation of all k-cell potential sub-
tours, the process moves to sub-tours of size (k + 1). For each sub-tour, all its children
are generated, and the process of Checking primitivity/Adding sub-tour elimination
constraint/Generating children is restarted.

The pseudo-codes of the algorithm of checking for primitivity and expanding the
elimination constraints are shown Appendix C.

3.7.5 Sub-Tour Elimination

Elimination Constraints

To avoid formation of a primitive sub-tour, it is necessary and sufficient to have at
least one (which implies two) edge incident to the nodes on its boundary. To eliminate
sub-tour containing cells cs and cg in Figure 3.12(a), at least two of the darkened edges
must exist on the loop.

A well-known sub-tour elimination constraint developed by Dantzig, Fulkerson,
Johnson [DFJ, 1953] for the Traveling Salesman Problem is expressed as follow;

\[ \sum_{m, n \in s} X_{mn} \leq |s| - 1 \]

\[ X_{mn} \in \{0, 1\} \]

\[ 2 \leq |s| \leq |C| - 2 \]
Figure 3.12: Representation of a sub-tour elimination constraint on the primal and dual

where \( C \) is the set of cities, and \( s \) is any sub-set of cities. The binary variable \( X_{mn} \) is equal to 1 if the edge from city \( m \) to city \( n \) is included, and is 0 otherwise.

A slight modification of the DFJ sub-tour elimination constraint is implemented in our problem.

\[
\sum_{m \in R_s} \sum_{n \in \overline{R_s}} Y_{mn} + \sum_{n \in R_s} \sum_{m \in \overline{R_s}} Y_{nm} \geq 2 \quad \forall s \in S
\]  \hspace{1cm} (3.15)

\[R_s = \{n : n \in \mathcal{N}_c, c \in s\}\]

\[\overline{R_s} = \{n : n \in \mathcal{N}_c, c \in \overline{s}\}\]

\( R_s \) is the set of nodes on the cells forming a sub-tour \( s \), and \( \overline{R_s} \) is the complement set of \( R_s \), where \( S \) is the set of primitive sub-tours.

**Dual Representation**

To define the sub-tour elimination constraint in the dual domain, all nodes corresponding to the cells in a sub-tour are merged into a representative node. Those
nodes in $G_x$ that are adjacent to any node in the sub-tour are assumed adjacent to the representative node. They form a star with the representative node. For all nodes $c, k$ on the star and adjacent in $G$, the edge $ck$ is added to the star. A pseudo-wheel appears. It defines the sub-tour elimination constraint in the dual domain. By a pseudo-wheel we mean a wheel that could have some non-degree 3 nodes on its rim Figure 3.13(a) \(^3\).

The sum of the binary variables corresponding to the edges on the most internal rim as well as any edge connected to a degree 2 node must be greater than or equal to 2. This is shown for the sub-tour containing cells $c_5$ and $c_6$ in Figure 3.12(b).

In this example the smallest pseudo-wheel is the only pseudo wheel, and is a wheel.

**Expanding the Sub-tour Elimination Constraints**

The elimination constraints of a considerable number of primitive sub-tours are expanded and become identical.

**Theorem 3** *Given the set of primitive sub-tours $S = \{ s : |s| \leq |r| \}$. There is a dominating sub-set $S' \subseteq S$, such that for any sub-tour $s \in S \setminus S'$, there exists a sub-tour $s' \in S'$ such that the elimination constraint of $s'$ is stronger than that of $s$."

**Proof:** We show that the pseudo-wheel corresponding to the elimination constraint of any sub-tour is expanded to its largest wheel. The pseudo-wheel of any sub-tour may contain two patterns violating the definition of a wheel.

Pattern 1: There are one or more degree 2 nodes. The internal node of edge $e_1$ in Figure 3.13(a) is an example of this situation.

Pattern 2: There are one or more degree 3 nodes inside the largest rim. This is the case for the node formed by edges $e_2$ and $e_3$ in Figure 3.13 (a).

Recall that the sub-tour elimination constraint in the dual corresponds to the edges on the smallest rim and any edge of pattern 1. By adding any degree 2 node

---

\(^3\)The edges that form the star are identified using adjacencies in $G_x$. Additional edges to form the pseudo-wheel are identified based on adjacencies in $G$. 

---
Figure 3.13: Expanding procedure for sub-tour elimination constraint
of the pseudo-wheel to the sub-tour, a new sub-tour is formed. This is because all nodes of the pseudo-wheel, including degree 2 nodes are adjacent to the sub-tour. The elimination constraint of the new sub-tour is stronger, because its (pseudo) wheel contains one less edge. Therefore, the elimination constraint of any sub-tour is expanded to its smallest wheel.

By analyzing pattern 2, we show that the smallest wheel is expanded to the largest wheel. The binary variable of any edge in a face in the dual graph is less than or equal to the sum of the binary variables corresponding to the remaining edges of that face. This is the content of constraint 3.9 in formulation-3. Therefore, having an edge replacing the remaining edges of its face always makes the corresponding constraint stronger. This is the case for replacing edges \( e_2 \) and \( e_3 \) by \( e_4 \). Therefore, the elimination constraint of any sub-tour is expanded to its largest wheel.

In both patterns, expanding the constraint does not cut any feasible portion of the search space. First, the expanded sub-tour may not become a complete tour because it does not cover any new node not already covered by the pseudo-wheel. Second, the expanded constraint corresponds to a specific sub-tour which nevertheless its eliminated constraint must be included in the model.

**Corollary 1** Given a primitive sub-tour \( s \), if \( G_s[\overline{s}] \) is not connected, then the largest wheel is expanded to a segment of it.

This is explained in Figure 3.14 for the primitive sub-tour \( s = \{c_6, c_b\} \). The edges of the largest wheel of this sub-tour are \( a, b, c, d, e, f, g, h \). However, either \( a, b, c, d \) or \( e, f, g, h \) are enough to eliminate this sub-tour.

In Figure 3.13(d), there are three nodes inside the largest wheel. Therefore the elimination constraint corresponding to the largest wheel could eliminate up to seven additional sub-tours. Since situation 1 exists between \( c_2 \) and \( c_4 \), a total of five additional sub-tours are eliminated.

As explained earlier, sub-tours with no edge on the boundary are all assumed primitive. For the purpose of computational efficiency, the expanding algorithm for
Figure 3.14: Expanding a largest wheel to a segment of it
Table 3.2: The number of potential and primitive sub-tours in a block layout

<table>
<thead>
<tr>
<th>Sub-tours not adjacent to the boundary</th>
<th>Number of cells block layouts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Average number of potential sub-tours in a block layout</td>
<td>244</td>
</tr>
<tr>
<td>Average number of primitive sub-tours in a block layout</td>
<td>16</td>
</tr>
</tbody>
</table>

Sub-tours not adjacent to the boundary is slightly different and less comprehensive than that of others.

Regarding these sub-tours, for any cell $c \in s$, if there is not any cell $k \in s$), such that $ck \in A$, then $s \rightarrow s \cup c$. This is the case for cell $c_{14}$ and the sub-tour containing cells $c_5$ and $c_6$ in Figure 3.13(b). Sub-tour $c_5c_6$ is expanded to sub-tour $c_5c_6c_{14}$, and the number of sub-tour elimination constraints is reduced by one.

Sub-tours adjacent to the boundary are expanded when checked for primitivity. The following algorithm expands primitive sub-tours.

**Expanding Procedure 1**

0) Given $s$, a primitive sub-tour, and $r$ the smallest component of $s$
1) Find $\overline{r}$ = The full complement of $r$, as the expanded sub-tour of $s$.
2) If $|r| < |\overline{r}|$, then $r$ is the expanded sub-tour.

Table 3.2 shows the number of potential and primitive sub-tours in layouts of size 10 to 40 cells. Ten layouts of each specific size were generated. The number of potential sub-tours for the block layout were counted up to and including $Min\{10, |C|/2\}$-cell sub-tours.

The average number of primitive sub-tours for 10 layouts of size 40 cells is shown in Figure 3.15. The maximum size of the sub-tours are shown on the horizontal line. The total number of primitive sub-tours of up to a specific size are shown on the vertical line.

Let's summarize the concepts developed in this section. Given any sub-tour $s$, all sub-sets $s' \subset s$ for which $s \subset s'$, and all superset $s'' \supset s$ for which $s'' \subset s$ require
Figure 3.15: The average number of up to $k$-cell sub-tours in 40 cell layouts
a unique elimination constraint. This is the constraint corresponding to the largest super-set.

### 3.7.6 Horizontal/Vertical Cuts

The block layout with the exact length of its edges is embedded in a plane. The faces are all right angle. These properties allow us to define a special set of cuts in the framework of DFJ constraints.

Suppose nodes of the block layout are sorted in increasing order in \( x \) (\( y \)) directions, and duplications are removed. The sets of sorted nodes are shown by \( x_1^o, x_2^o, \ldots, x_{NX}^o \) and \( y_1^o, y_2^o, \ldots, y_{NY}^o \), where \( NX \) and \( NY \) are the total number of non-repeated coordinates in \( x \) and \( y \) directions respectively. A set of vertical cuts are defined at points \( x_i^o + \varepsilon \), for \( i = 2 \) to \( NX - 2 \). Similarly, a set of horizontal cuts are defined at points \( y_i^o + \varepsilon \), for \( i = 2 \) to \( NY - 2 \). These cuts are easily identified, and their number is linear in the number of cells. In the layout of Figure 3.16, there are 4 horizontal and two vertical cuts and constraints. The cut constraint requires the loop to have at least two of its edges intersecting the cut. These cuts do not eliminate those sub-tours overlapping their complement in both \( x \) and \( y \) directions, Figure 3.17.

**Property 8** A horizontal/vertical cut constraint eliminates a set of primitive sub-tours. The cut constraint is stronger than the elimination constraints of the corresponding sub-tours.

Proof: A cut splits the graph into two components, say \( G_1(C_1, A_1) \) and \( G_2(C_2, A_2) \). Sub-tours \( C_1 \) and \( C_2 \) are eliminated by the cut.

Given \( s_{cut} \) as the set of cells on the cut, any sub-tour \( s_1' \subset C_1 \) as long as \( s_{cut} \subset s_1' \) is also eliminated by the cut constraint. Indeed, all sub-tours not having overlap with their complement in either \( x \) or \( y \) direction are not primitive. They are all eliminated by the cut constraints. The vertical cut in Figure 3.18 eliminates 7 potential sub-tours. If instead of one cell, there were \( n \) cells in the block of cell \( F \), then at least \( n + 6 \)
Figure 3.16: Horizontal and Vertical CUTS for the 11-cells example
sub-tours were eliminated by this cut. We show that the cut constraint is stronger than the elimination constraint of Figure 3.18(g). Elimination constraints of all other sub-tours are special cases of this one.

The loop configuration implies the value of the variable of each edge to be equal or less than some of the remaining defining edges of any of its nodes. Therefore

\[ a + b \geq d \]  \hspace{1cm} (3.16)

and it implies

\[ a + b + c \geq e \]  \hspace{1cm} (3.17)

Given any sub-tour elimination constraint, the value of each of its binary variables is equal or less than sum of the values of the remaining variable. Therefore, regarding the elimination constraint of sub-tour \( F \), the following holds

\[ c + d \geq e \]  \hspace{1cm} (3.18)

No matter whether the elimination constraint of sub-tour \( F \) is included in the model or not, the above inequality is a valid one. Furthermore, even if the block of cell \( F \) were occupied by a large number of cells, still only one of the edges of the corresponding
Table 3.3: Sub-tour elimination constraints required for the 11-cell example

A loop configuration implies the value of the variable of each edge to be equal or less than some of the remaining defining edges of any of its nodes. Therefore

\[ a + b \geq d \]  

(3.19)

and it implies

\[ a + b + c \geq e \]  

(3.20)

The same proof is applied to \( C'_2 \subseteq C_2 \). However, in the example of Figure 3.18, \( s'_2 = \emptyset \), all of the sub-tours on the right hand side of the cut have \( x \) direction overlap with their complement.

In small problems, say 10-15 cell problems, a considerable number of sub-tours are eliminated by these cuts. As Table 3.3 shows, more than half of the potential sub-tours of the 11-cell example are eliminated by horizontal/vertical cuts, and more than 85 percent by horizontal/vertical cuts and 1-cell sub-tour elimination constraints.
Figure 3.18: A CUT constraint eliminates a set of sub-tours.
3.8 Computational Consideration

3.8.1 The Shortest Loop Model

Now assume the existence of a linear cost function that can be adequately utilized as a measure of the effectiveness of the model. When the cost of constructing and maintaining an edge is proportional to its length, the shortest loop problem is stated as

\[
\text{Min} \quad Z = \sum_{mn \in A} Y_{mn}l_{mn} \quad (3.21)
\]

\[
\sum_{mn \in Y_c} Y_{mn} \geq 1 \quad \forall c \in C \quad (3.22)
\]

\[
\sum_{m \leq n} Y_{mn} + \sum_{n \leq k} Y_{nk} \leq 2 \quad \forall n \in N \quad (3.23)
\]

\[
\sum_{i < m} Y_{im} + \sum_{h > m} Y_{mi} \geq Y_{mn} \quad \forall mn \in A \quad (3.24)
\]

\[
\sum_{i < n} Y_{in} + \sum_{n \neq i} Y_{ni} \geq Y_{mn} \quad \forall mn \in A
\]

\[
\sum_{m \in R, n \in \overline{R}} Y_{mn} + \sum_{n \in R, m \in \overline{R}} Y_{nm} \geq 2 \quad \forall s \in S \quad (3.25)
\]

\[
Y_{mn} \in \{0, 1\} \quad \forall mn \in A \quad (3.26)
\]

If the fixed and variable costs of construction and operations differ among edges of the loop, then a cost coefficient is included in the objective function

\[
\text{Min} \quad Z = \sum_{mn \in A} Q^e_{mn}l_{mn}Y_{mn} \quad (3.27)
\]

\[Q^e_{mn}\] is the total fixed and variable costs allocated to each unit length of edge \(mn\).

The model was applied on a set of test problems. Test problems were generated using a set of layouts in Apple [1977], Francis et al. [1992], Sinriech and Tanchoco [1993], Tompkins et al. [1996]. We slightly modified them, such that each layout contained 10 cells. To form a 20-cell layout, two 10 cell layouts were randomly selected, one side
A Loop Material Flow System Design for Manufacturing Plants

<table>
<thead>
<tr>
<th>Number of the cells in the layout</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average CPU time (seconds)</td>
<td>.08</td>
<td>.5</td>
<td>2.6</td>
<td>9.3</td>
</tr>
<tr>
<td>Min</td>
<td>.05</td>
<td>.27</td>
<td>.8</td>
<td>1.85</td>
</tr>
<tr>
<td>Max</td>
<td>.1</td>
<td>.7</td>
<td>7.3</td>
<td>13.33</td>
</tr>
<tr>
<td>Average number of iterations</td>
<td>46</td>
<td>128</td>
<td>307</td>
<td>381</td>
</tr>
<tr>
<td>Min</td>
<td>34</td>
<td>99</td>
<td>184</td>
<td>259</td>
</tr>
<tr>
<td>Max</td>
<td>57</td>
<td>143</td>
<td>632</td>
<td>532</td>
</tr>
<tr>
<td>Average number of nodes</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Min</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Max</td>
<td>0</td>
<td>13</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.4: CPU times, number of iterations, and nodes for shortest loop problem of layouts of sizes 10-40 cells

of each layout was randomly selected as the interface of the pair of layouts. 40-cell layouts were formed by random selection of four 10-cell layouts, and random selection of some of their sides. The layouts were jointed by the selected sides.

The problems were solved using CPLEX on a SUNstation Model 20. CPU times, the number of nodes and iterations are summarized in Table 3.4. The optimal solution of our model for a considerable number of problems reported in table 3.4 were found in the root node of the branch and bound. However, the computational times required to find the primitive sub-tours are not included. The CPU times required to identify up to and including 8-cell sub-tours for 10, 20, 30, and 40 cell layout was 2, 54, 615, and 2070 seconds respectively. However these are times required by our not very efficient routines and are not comparable with those of extremely efficient routines of CPLEX. Our computational experiments on the reported 10-40 cell layouts indicate that by including the elimination constraint of the small size sub-tours, say sub-tours of up to size 4 in a 40-cell layout, a single loop emerges. The tendency towards forming sub-tours in practical layouts does not seem strong.

The CPU times required to identify up to 4-cell sub-tours for 10, 20, 30, and 40 cell layout was 0.5, 1.5, 9, and 24 seconds respectively.
3.8.2 Earlier Formulations

Afentakis [1989] developed a loop layout for flexible manufacturing systems. The concept of designing a single loop for Automated Guided Vehicle Systems was proposed by Bartholdi and Platzman [1989]. Kouvelis and Kim [1992], designed a uni-directional loop layout for automated manufacturing systems. Designing a single loop on the edges of a fixed block layout was initiated by Tanchoco and Sinriech [1992]. Their formulation stated in the framework of our notation is summarized below;

Notations

\[
X_{mn} = \begin{cases} 
1 & \text{if the directed arc } mn \text{ is included in the loop} \\
0 & \text{otherwise}
\end{cases}
\]

\[
z_n = \begin{cases} 
1 & \text{if node } n \text{ is included in the loop} \\
0 & \text{otherwise}
\end{cases}
\]

\[
\text{Minimize } \sum_{mn \in \tilde{A}} l_{mn} X_{mn}
\]

\[
\sum_{mn \in \mathcal{E}_c} X_{mn} + X_{nm} \geq 1 \quad \forall c \in C
\]

\[
\sum_{mn \in \tilde{A}} X_{nm} = z_n \quad \forall n \in N
\]

\[
\sum_{mn \in \tilde{A}} X_{mn} = z_n \quad \forall n \in N
\]

\[
X_{mn} \in \{0, 1\} \quad \forall mn \in \tilde{A}
\]

The constraint derived by Miller, Tucker, and Zemlin [MTZ, 1960] was implemented for sub-tour elimination.

\[
u_m - u_n + |N|X_{mn} \leq |N| - 1 \quad \forall mn \in \tilde{A}
\]

\( u_m \) is the stage of the travel at which city \( m \) is visited.

Our model differs from the above model in the following aspects.

1) Smaller number of binary variables: their model needs twice the number of binary variables as ours.
2) A different sub-tour elimination approach with a manageable number of constraints: The number of constraints is the main advantage of the MTZ approach. It is linear in the number of cells. This is the main disadvantage of DFJ constraints. It grows exponentially in the number of cells. However, as shown in Table 3.2 the number of DFJ constraints for practical layouts is not large. The average number of the MTZ constraints was 18 and 30 for ten and twenty cell test problems respectively. The number of DFJ constraints was 16 and 111 respectively.

3) Stronger LP-Relaxation: MTZ constraints are weak and play no role in lifting the LP-Relaxation. Indeed, no other sub-tour elimination constraint has a stronger LP-relaxation than DFJ, Laporte [1992].

4) Direct: The MTZ becomes much stronger if one modifies it to

\[ u_m - u_n + |N| (X_{mn} + X_{nm}) - 2X_{nm} \leq |N| - 1 \quad mn \in A \]

However, the model is still unable to find the shortest loop directly. It requires a node to be duplicated. The duplicated node is assumed as the first and the last city of the travel. It is automatically included in any feasible and optimal solution. This duplication does not create any difficulty in the TSP because all nodes are covered by a feasible (optimal) solution. This is not the case in our problem. It is not known in advance which nodes will be on the optimal loop. Fixing any node in the solution, immediately cuts off a portion of the search space, a portion which may include the optimal solution. Therefore, the final solution of this approach is not guaranteed to be the global optimal any more. If cell \( c \) has the minimum number of nodes on its boundary, then the formulation must be repeated for \((J_c - 1)\) nodes. By comparing their final solutions, the overall shortest loop is obtained. Our formulation directly finds the shortest loop.

### 3.8.3 A Goal Programming Orientation

In block layouts, a considerable number of edges have the same length. Therefore many solutions have the same objective function value. This increases Branch and
Figure 3.19: Multiple optimal solutions of the example

Bound computations since a larger number of nodes have to be examined. Since the solution time for practical layouts are low, and usually there are multiple optimal solutions, there is enough motivation to approach the problem in a goal programming orientation.

Some requirements of the real life flow patterns may be incorporated in the model. These are; a) The shortest loop with the smallest number of edges. b) The shortest loop with the sum of degrees of its covered nodes maximized. c) The shortest loop with the minimum number of corner points.

Case (a) is when the variable cost which is proportional to the length of the edges is the same for all edges, while for opening each edge a high fixed cost independent of the length of the loop is also incurred. In this case, the shortest loops in Figure 3.19(a),(3) are preferred.

The physical interpretation of case (b) is to have more access points to the cells. If the weights of 3 and 4 are assigned to the nodes of corresponding degrees, then Figure 3.19(e) is the best shortest loop.
Physical interpretation of case (c) is to avoid excessive turns. In this case, for all degree 3 nodes \(n\), the term \(2Y_{mn} - Y_{in} - Y_{jn}\) form a term of the secondary objective function. Where, \(mn, in, jn\) are edges incident to \(n\), and \(in, jn\) are in the same direction. For all degree 4 nodes \(n\), where the edges incident to it are say \(in, jn\) in one direction, and \(mn, on\), in the other direction, the following constraints are added to the model.

\[
y_{mo}^+ - y_{mo}^- = Y_{mn} - Y_{on}
\]
\[
y_{ij}^+ - y_{ij}^- = Y_{in} - Y_{jn}
\]

The term \(y_{mo}^+ + y_{mo}^- + y_{ij}^+ + y_{ij}^-\) is the other term in the secondary objective function. Figure 3.19(a) is the optimal solution for this situation.

Sometimes it may be preferable not to have the material handling system adjacent to the walls of the building. Both for the difficulty of the installation, and lack of flexibility to access the system from both sides. In this case, a constraint to have the sum of the edges on the external cell equal to one is added to the model. The one edge is the receiving/shipping point for the manufacturing facility.

### 3.9 Conclusion

This chapter was devoted to a problem with some similarities to the traveling salesman problem. The properties of the problem were discussed. Based on these properties, a formulation with the smallest number of variables, strong constraints, and manageable number of sub-tour elimination constraints was developed. Computational results, at least for practical size layouts are promising.

In the next chapter, we add material flow to the problem and extend the model to trip-based material handling systems.
Chapter 4

A Loop Material Flow System Design for Automated Guided Vehicle System

4.1 Introduction

In this chapter we develop a uni-directional single loop material flow system for Automated Guided Vehicles. The concept of designing a uni-directional single loop on a fixed block layout was initiated by Tanchoco and Sinriech [1992]. Their procedure contains 5 phases.

Phase-1 employs an integer program to find a loop covering at least one arc of every cell. More than one run is required to obtain the shortest loop. Phase-2 includes an enumeration process to explore all feasible and infeasible loops. Phase-3 applies a set of 3 rules on all possible feasible and infeasible loops identified in phase-2 and eliminates inferior loops. In phase-4 another mixed integer programming model is applied on the remaining loops to find P/D station locations. In Phase-5, all solutions of phase-4 are compared and the overall optimal loop to minimize the loaded vehicle travel is identified. Sinriech and Tanchoco [1993] propose a heuristic for phase-1, and a preprocessing routine for phase-3.

In this chapter we develop an exact formulation to design a loop material flow system for Automated Guided Vehicles. Edges and nodes of a fixed block layout
are candidates for flow path and pick-up and delivery station locations. The model integrates both the design of the uni-directional loop material flow pattern, and determination of the number and location of stations.

Two objective functions are examined. One is the loaded vehicles travel distance. The other is the total cost of the system including the loop, P/D stations, and vehicle travel. Some extensions to other configurations are discussed.

We explain our formulation and computational results, using an example by Sinриech and Tanchoco, [1993]. The block layout and its From-To chart are shown in Figure 4.1 and Table 4.1. ¹

Our work differs from that of others in the following aspects
1) A smaller number of binary variables is used to find the loop.
2) The shortest loop is found directly.
3) Finding the loop and the location of Pick-up and Delivery stations are integrated. The optimal solution is identified in a single run.
4) The model is extended to multiple stations per cell. Unlike previous models, solution time for multiple station per cell is much less than that of single station.
5) The importance of tightening the feasible region of the LP-relaxation and integer solutions is discussed. Achieving a tight LP-relaxation at the expense of increasing the number of constraints and preprocessing is justified. A Branch and Bound algorithm based on the properties of the problem is proposed.
6) In addition to the minimization of the loaded vehicle travel, a cost based model is also formulated. The objective function includes the total installation and other fixed and variable costs of the loop, P/D stations, and loaded vehicle travel.
7) The model is extended to bi-directional loop conventional configuration.

In section 4.2 we develop the mathematical model. Theoretical foundations of the relationships between LP-Relaxation and IP solutions are discussed in sections 4.3

¹Note that the external cell could also have both inflow and outflow. The outflow of the external cell is the total material received by the manufacturing plant. The inflow of the external cell is the total finished product shipped from the manufacturing plant.
Figure 4.1: The layout of the example from Sinriech and Tanchoco [1993]

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>M</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>80</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>80</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>10</td>
<td>65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>80</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>15</td>
<td>25</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>80</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>10</td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>J</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>55</td>
</tr>
<tr>
<td>K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>Total</td>
<td>80</td>
<td>10</td>
<td>80</td>
<td>10</td>
<td>30</td>
<td>20</td>
<td>50</td>
<td>55</td>
<td>70</td>
<td>80</td>
<td>485</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: FT-chart for the example of Figure 1

and 4.4. Computational results are reported in section 4.5. Extensions are formulated in section 4.6. Directions for future research are outlined in section 4.7.

## 4.2 Model Formulation

### 4.2.1 Formal Statement

Let us define our Flow System Design Problem using the planar graph $G_b(N, \bar{A})$ associated with the block layout. $N$ is the set of intersections on the boundaries of cells, $\bar{A}$ is the set of directed arcs. For any adjacent pair of nodes in $N = \{n_1, \ldots, n_{|N|}\}$,
there exists two directed arcs, \( n_i, n_j \) and \( n_j, n_i \), where \( i \neq j \) and \( n_i, n_j \in N \). The graph is embedded in a plane. Arcs are either horizontal or vertical, therefore, the maximum degree of a node is 8. Feasible Loop is a circuit containing at least one arc of each cell.

A distance function \( l : A \rightarrow \mathbb{R}^+ \), assigns \( l_{n_i,n_j} \) as the length of the edge connecting nodes \( n_i \) and \( n_j \). The length of each arc is equal to the rectilinear distance between its nodes.

There is a (usually non-planar) graph \( G_f(C, A_f) \) representing the material. For each pair of cells \( c, k \) with a strictly positive flow from \( c \) to \( k \), there is a directed arc in \( A_f \) connecting \( c \) to \( k \). A function \( f : A_f \rightarrow \mathbb{R}^+ \), assigns \( f_{ck} \) as the intensity of material flow to the arc \( ck \in A_f \).

Nodes on the boundaries of each face are candidates for station location. One \( P \) and one \( D \) station is allowed per cell, the stations are not necessarily combined.

Given \( t_{cmn} \) as the intensity of the outflow of cell \( c \) passing arc \( mn \). The Loaded Flow System Design Problem is to find a uni-directional loop and its station locations, such that the loaded vehicle travel distance is minimized. That is to minimize sum of the \( (t_{cmn} \times l_{mn}) \) over all cells.

In the Cost Flow System Design Problem, each cell is allowed to have more than one \( P \) and \( D \) stations. There is a cost associated with construction of each arc and station, as well as with the movement of loads through the loop. The problem is to find a loop which minimizes the total cost of the system.

The problem has both Location/Allocation and Generalized Traveling Salesman elements.

### 4.2.2 Loop Flow Pattern

**Formulation-1**

A binary variable \( X_{mn} \) is defined for the directed arc \( mn \). The variable is equal to 1 if the directed arc is on the loop, and 0 otherwise. The formulation is stated as follow.
Each cell has at least one arc in the final loop.

\[ \sum_{m,n \in \mathcal{N}_c} X_{mn} \geq 1 \quad \forall c \in C \quad (4.1) \]

Arcs are uni-directional

\[ X_{mn} + X_{nm} \leq 1 \quad \forall mn \in A \quad (4.2) \]

At most one incoming arc to each node.

\[ \sum_{m \in \mathcal{A}} X_{mn} \leq 1 \quad \forall n \in N \quad (4.3) \]

\( \mathcal{A} \) is the set of directed arcs. The number of incoming arcs at each node is equal to the number of outgoing arcs.

\[ \sum_{mn \in \mathcal{A}} X_{mn} = \sum_{nm \in \mathcal{A}} X_{nm} \quad \forall n \in N \quad (4.4) \]

\[ X_{mn} \in \{0, 1\} \quad \forall mn \in \mathcal{A} \quad (4.5) \]

This formulation is adopted from Sinriech and Tanchoco [1993].

**Formulation-2**

Since the distance matrix is symmetric, therefore, instead of two binary variables, one for each direction of arcs, only one binary variable is defined for the corresponding nondirected edge. The binary variable \( Y_{mn} \) where \( n > m \) is equal to 1, if the nondirected edge \( mn \) is on the loop, and 0 otherwise. A set of binary variables are also required for the nodes on the layout. The binary variable \( u_n \) is equal to 1, if node \( n \) is on the loop, and is 0 otherwise.

Each cell has at least one edge on the loop.

\[ \sum_{mn \in \mathcal{X}_c} Y_{mn} \geq 1 \quad \forall c \in C \quad (4.6) \]
Nodes are either of degree 0 or 2.

\[ \sum_{m<n} Y_{mn} + \sum_{k>n} Y_{nk} = 2u_n \quad \forall n \in N \quad (4.7) \]

This formulation is adopted from the Generalized Traveling Salesman Problem formulation by Laporte and Nubert [1983].

**Formulation-3**

Although the cells of the block layout need not be convex, they are all right angle polygons. The degree of nodes is therefore restricted to 4. This property, coupled with the symmetry of the distance matrix, is utilized to reduce the number of binary variables and to formulate the degree 2 constraint.

As in Formulation 2 only one binary variable is used for each nondirected edge.

No binary variable is required for nodes.

Equation 4.6 remains unchanged.

At most 2 edges incident to a node are on the loop.

\[ \sum_{m<n} Y_{mn} + \sum_{n<k} Y_{nk} \leq 2 \quad \forall n \in N \quad (4.8) \]

No node only have one edge on the loop. In other words, at each degree \( k \) node, the sum of the decision variables corresponding to each set of \( k - 1 \) edges is greater than or equal to the remaining one.

\[ \sum_{i<m} Y_{im} + \sum_{i>n} Y_{mi} \geq Y_{mn} \quad \forall mn \in A \quad (4.9) \]

\[ \sum_{i<n} Y_{in} + \sum_{i>n} Y_{ni} \geq Y_{mn} \quad \forall mn \in A \]

\[ Y_{mn} \in \{0,1\} \quad \forall n > m \]

The number of these constraints for a block layout is between \( 3|N| \) to \( 4|N| \). If each cell on the average contains 5 edges, then \( N \) is almost \( 1.5|C| \). Therefore, the number of these constraints in our problem is less than \( 6|C| \), compared to \(|C|^2 \) in TSP.
When the objective is to find the shortest loop, the direction is not important. It could simply be directed in one direction or the other. However, for the loaded vehicle travel objective function, the direction of the loop is a required part of the optimal solution. In the presence of material flow, formulation-2 is unable to model the problem. Additional binary variables are required for the second direction of the edges. By assuming a few $X_{mn}$ variables as integers, formulation-3 directs the loop. Therefore when material flow is added to the problem, the number of binary variables in the three formulations are $2|A|$, $2|A| + |N|$, and $|A| + |c|$ respectively. $c$ is a cell with the minimum number of edges.

Unidirectionality in formulation-3 is achieved using the following constraint

$$X_{mn} + X_{nm} = Y_{mn} \quad \forall mn \in A$$

All $0 \leq X_{mn} \leq 1$ are real variables, while their corresponding $Y_{mn}$ is integer. The only exceptions are the edges on the boundary of cell $c$. Its $Y_{mn}$s are left real, while the corresponding $X_{mn}$s are integers. Directed arcs of this cell when coupled with the arc balance constraint, play the interface role and design a uni-directional loop. In other words, after branching on edge and arc integer variables, the LP-relaxation of the problem coincides with IP solutions with respect to the remaining arcs.

The "arc balance constraint" constraint 4.4 in Formulation-1 is repeated for Formulation-3. However, most $X_{mn}$s, are real variables in the later formulation.

A real variable $V_n$ which comes out 0 or 1, indicates whether node $n$ is on the loop or not.

$$V_n = \sum_{mn \in A} (X_{mn} + X_{nm})/2 \quad \forall n \in N$$

Formulation-3 with the above modifications is implemented in the rest of this chapter.

**Number and location of Stations**

Nodes on the boundary of each cell are candidate locations for its P/D stations. Given $\mathcal{N}_c$ as the set of nodes on cell $c$, a pair of binary variables are defined for each node.
The binary decision variable $P_{cn}$ is equal to 1 if node $n$ is selected as the pick-up station of cell $c$, and 0 otherwise. Similarly, the binary decision variable $D_{cn}$ is equal to 1 if node $n$ is selected as the delivery station of cell $c$, and 0 otherwise.

By inserting a set of new degree 2 nodes on the edges, the model is extended to the case of stations on edges. As we will see later, the LP-relaxation of the station location variables is close to IP solutions. Increasing the number of candidate station locations does not increase the solution time substantially. The model is first restricted to one P and one D station per cell, later this restriction is relaxed.

$$\sum_{n \in \mathcal{N}_c} P_{cn} = 1 \quad \forall c \in C$$
$$\sum_{n \in \mathcal{N}_c} D_{cn} = 1 \quad \forall c \in C$$
$$P_{cn}, D_{cn} \in \{0, 1\} \quad \forall n \in \mathcal{N}_c \quad \forall c \in C$$

Therefore, on the average sum of

$$\sum_{c \in C} \mathcal{N}_c/|C|$$

station location variables is equal to 1. The number of these variables in our example is 86. On the average every 4.3 variables has a covering equality of 1. A considerable number of these variables do not require branching. They automatically come out 0 or 1, i.e., they could be removed from the set of integer variables.

### 4.2.3 Material Flow

A multi-commodity material flow is transferred through the proposed loop. We initially assume the outflow of each cell, i.e., the total outflow of each row of FT-chart, to be a commodity, other possibilities are discussed later. The decision variable $t_{ckmn}$ shows the intensity of flow from cell $c$ to cell $k$ on arc $mn$. The notation $t_{cmn}$ is used for the intensity of the total outflow of cell $c$ on arc $mn$. 
To the best of our knowledge, ours is the first model for simultaneous design of a loop configuration and determination of the number and location of pick-up and delivery stations.

Material Transfer along Arcs and Nodes

Material flow is only through the arcs on the loop.

\[ \sum_{c \in C} t_{cmn} \leq M X_{mn} \quad \forall mn \in \tilde{A} \quad (4.14) \]

where \( M \) is a large number. We refer to this constraint as the Flow Feasibility constraint.

There may exist a pair of adjacent cells with all their flows between themselves. The following constraint is required for the common nodes of such pairs of cells.

\[ \sum_{c \in C_n} (P_{cn} + D_{cn}) \leq M \sum_{mn \in \tilde{A}} Y_{mn} \quad \forall n \in N \quad (4.15) \]

\( C_n \) is the set of cells adjacent to node \( n \). This constraint is automatically satisfied for other cells through the flow balance constraint.

It is usually preferred to avoid an unrealistic optimal solution in which many P/D stations are clustered together. Furthermore, it is not likely to use AGVs as a stationary material transfer device. Such moves are carried out either by a pick-and-place robot, the human operator, a short gravity or powered conveyor, or a ball transfer. Therefore, a limit may be imposed on the total number of P/D stations or total load transferred between adjacent cells at some nodes. In this case, the following constraints are added to the model.

\[ \sum_{c \in C_n} (P_{cn} + D_{cn}) \leq U S_n \quad \forall n \in N \quad (4.16) \]

\[ \sum_{c \in C_n} (P_{cn} f_c + D_{cn} \sum_{k \in A_f} f_{kc}) \leq U F_n \quad \forall n \in N \quad (4.17) \]
$f_{kc}$ and $f_c$ are the inflow to cell $c$ from cell $k$, and the total outflow of cell $c$ respectively. $US_n$ is the maximum number of stations at node $n$, while $UF_n$ is the maximum flow transferred through node $n$. These constraints increase the solution times. Alternatively, by locating stations on the edges, no more than 4 stations are assigned to a node, and the maximum flow is restricted to the total flow between a pair of adjacent cells.

**Material Flow Balance**

The multi-commodity flow balance constraints state that; for each specific flow, the total inflow to a node from all nodes and cells adjacent to it is equal to the total outflow from the node to the adjacent nodes and cells. This constraint for the case of assuming the total outflow of each cell as a commodity is

$$f_cP_n + \sum_{mn \in A} t_{cmn} = \sum_{ck \in A_f} f_{ck} D_{kn} + \sum_{nm \in \overline{A}} t_{cnm} \quad \forall n \in N \quad \forall c \in C \quad (4.18)$$

Node $n$ could be a supply point, a demand point, both supply and demand point, or a transshipment point. Constraints 4.18, 4.11, 4.12 ensure that the total commodity sent out from each cell is equal to the flow of its corresponding row in the FT-chart. The total material sent from an origin cell is equal to the amount received by all its destinations.

**4.2.4 Sub-Tour Elimination**

The DFJ sub-tour elimination constraint, with a small modification is adopted to our model as follows

$$\sum_{i \in R_s} \sum_{j \in \overline{R_s}} X_{ij} \geq 1 \quad \forall s \in S \quad (4.19)$$

Where $S$ is the set of primitive sub-tours explained in chapter 3. $R_s$ is the set of nodes on the cells forming sub-tour $s$, and $\overline{R_s}$ is the set of the remaining nodes. Miller, Tucker, Zemlin sub-tour elimination constraint has been implemented in previous formulations for designing a loop flow path on a block layout. The main advantage of
this constraint is its small number, but there are two difficulties. It requires a node
duplication, therefore more than one run is required to identify a loop in the block
layout. DFJ constraints directly find a loop, or the shortest loop. The difficulty DFJ
approach is its huge number of constraints. However, compared to TSP, this number
is much less in the block layout problem. As discussed in chapter 3, the average
number of up to 10 cell sub-tours are 16, 111, 771, 3635 for 10, 20, 30, 40 cell test
problems.

Furthermore, the Flow Feasibility Constraint 4.14 coupled with the flow balance
constraint 4.18 play the role of sub-tour elimination constraint. In other words, for a
majority of primitive sub-tours, while their elimination constraints are not explicitly
stated in the model, they are implicitly enforced by the material flow constraints. A
small number of DFJ constraints are enough to eliminate all sub-tours of a practical
layout.

Recall our discussion on primitive sub-tours in chapter 3. Since there was no
material flow in the system, most of the 1-cell sub-tours were primitive. In the 11-cell
example, 1-cell sub-tours of D, J, F, M are primitive. When material flow is added to
the model, none of the above primitive sub-tours requires an elimination constraint.

If a cell has a flow with a non-adjacent cell, then the flow feasibility and flow
balance constraints guarantee to have at least one edge on the loop connecting the
1-cell sub-tour to the non-adjacent cell. That is, the sub-tour elimination constraint is
implicitly enforced by the model. The justification remains true for larger sub-tours.

Given sub-tour $s$, suppose $g$ is the set of cells having one edge in common with $s,$
and $\overline{g} = C \setminus (s \cup g).$ If there is any cell in $g$ not adjacent to $\overline{g},$ removed it from $g$ and
add it to $s.$ Whenever there is a flow between $s$ and $\overline{g},$ or between two components
of $\overline{g}$ the sub-tour is no longer primitive.

Figure 4.2(a) is the adjacency graph of the prototype example excluding the ex-
ternal cell, Figure 4.2(b) is its flow graph. Suppose the sub-tour $s = \{F, H\}$ is under
consideration, the adjacency graph in Figure 4.2(c) shows, $s = \{C, D, J, M\},$ and
$\bar{s} = \{A, B, G, K, L\}$. Since $M$ is only adjacent to the cells in the sub-tour, it is removed from $\bar{s}$ and added to $s$. These partitions are shown on the flow graph in Figure 4.2(e). Flow edges connecting nodes on the same partition or component are removed from the flow graph. Since there are two flows between $s$ and $\bar{s}$ in Figure 4.2(f), therefore sub-tour $FH$ is automatically eliminated by material flow constraints.

When including the material flow in the model, the number of required sub-tour elimination constraints for the prototype example is reduced from 8 to 6 for 1-cell, from 9 to 4 for 2-cell, and from 4 to 2 for 3-cell sub-tours. All larger sub-tours are automatically eliminated by 1 to 3 cell sub-tour elimination constraint. The total number of DFJ constraints is around one percent of an 11-city TSP. However, the prototype example corresponds to a 12-city TSP.

It is also possible to ignore all sub-tour elimination constraints. If the flow graph is strongly connected, the flow path automatically comes out connected. If not, the flow pattern forms two or more disconnected loops, it could motivate a tandem configuration.

### 4.3 LP-Relaxation

Having an LP-relaxation tight to IP solutions is the most important factor in the solution time of Integer Programming models. It makes the value of the linear objective function close to its IP value. More accurate bounds are obtained, a majority of branches are on the right direction.

We tighten the feasible region of the LP-relaxation at the root and subsequent nodes.

1) At the root node: By preprocessing and increasing the number of constraints, to start from an initial LP solution close to IP values.

2) At subsequent nodes: By defining an intelligent taxonomy of variables for Branch and Bound. After each layer of branches, the LP-relaxation of the problem gets closer to IP solutions in the next layers.
A Loop Material Flow System Design for Manufacturing Plants

Figure 4.2: Graphical representation of the procedure to screen unnecessary primitive sub-tours.
In this section the theoretical foundations of the LP-Relaxation of our formulation are discussed. Based on this fundamental insight, the required number of binary variables, appropriate forms of constraints, their most efficient coefficients, and the taxonomy of variables for Branch and Bound are defined.

4.3.1 LP-Relaxation of P stations

Only the common nodes between an origin cell and its destinations motivate the origin cell to split its outflow among some of the candidate P stations.

**Theorem 4** Given cell c with no disconnected edges alongside the loop. For all nodes n on cell c but not on any of its destinations, the LP value for \( P_m \) is integer.

**Proof:** Suppose two nodes \( i, j \) are selected as the P stations of cell \( c \) in the LP-relaxation, i.e., \( 0 < P_{ci}, P_{cj} < 1 \). Without loss of generality, suppose \( j \) is located after \( i \) on the boundary of cell \( c \) on the directed loop. The contribution of cell \( c \) in the total outflows \( \times \) distance is

\[
O_c = P_{ca}O^{(i)}_c + P_{cj}O^{(j)}_c
\]  

(4.20)

\( O^{(i)}_c \) is the contribution of cell \( c \) in the total outflow \( \times \) distance when node \( i \) is selected as its P station. \( O_c \) is the contribution of cell \( c \) in the total outflow \( \times \) distance in the system. Since

\[
O^{(i)}_c = O^{(j)}_c + f_cL_{ij},
\]  

(4.21)

Therefore,

\[
O_c = O^{(j)}_c + P_{ca}f_cL_{ij}
\]  

(4.22)

\( L_{ij} \) is the length of the line segment on the directed loop from node \( i \) to node \( j \). The above expression is minimized, when \( P_{ci} = 0 \). Therefore, assigning any non-zero value to any node except to node \( i^* \) which is the last node of cell \( c \) on the directed loop, will increase the value of the linear objective function.

\[
P_{ca^*} = 1 \quad P_{ci} = 0 \quad i \neq i^*
\]
Therefore, for those nodes \( n \) on the origin cell \( c \) but not on any of its destinations, although \( P_{cn} \) is not among integers, it automatically come out zero or one.

**Corollary 2** In the LP-relaxation of the problem, the outflow of a cell \( c \) to its adjacent destinations is transferred through their common nodes. Therefore, some \( P_{cn} \) variables are fractional in the LP-relaxation.

Proof: Suppose there is a flow from cell \( c \) to its adjacent cell \( k \), and node \( i^* \) as explained in Theorem 4 is the \( P \) station of cell \( c \). By allocating each unit of the outflow of cell \( c \) to node \( i \) which is in common between \( c \) and \( k \), the linear objective function is decreased by \( L_{ii^*} \), and \( P_{ci} = f_{ck}/f_c \) decrease it by \( f_{ck} L_{ii^*} \). The variable \( P_{ci} \) does not pass this value unless node \( i \) is on the boundary of more than one destination. Otherwise, any additional increase in \( P_{ci} \) will increase the objective function by \( f_c L_{ii^*} (P_{ci} - f_{ck}/f_c) \).

Therefore, decision variables corresponding to common nodes between an origin and its destinations have to be considered as integer variables.

\[
P_{cn} \in \{0, 1\} \quad \forall n \in N_c \cap ( \bigcup_{c \in A_j} N_k) \quad \forall c \in C
\]

(4.23)

For example, all binary variables of \( P \) station locations of cell \( F \) automatically come out zero or one. For cell \( G \) it is enough to have \( P_{G4}, P_{G5} \) and \( P_{G8} \) among integer variables. The number of \( P \) station integer variables in the prototype example is reduced from 43 to 13.

**Corollary 3** If cell \( c \) with more than one destination has non-adjacent edges on the loop, then some \( P_{cn} \) variables come out fractional.

Proof: Suppose cell \( c \) has two disconnected segments on the loop, and their last nodes in the direction of the loop are \( i \) and \( j \). When node \( i \) is selected as a \( P \) station of cell \( c \), the total contribution of cell \( c \) in outflow \( \times \) distance in the system is

\[
O_c = \sum_{k \in Kc_{ij}} \sum_{mn \in A_{ij}} t_{ckmn} l_{mn} + L_{ij} \sum_{k \in Kc_{ij}} f_{ck} + \sum_{k \in Kc_{ij}} \sum_{mn \in A_{ji}} t_{ckmn} l_{mn}
\]
\[ A_{ij} = \{ mn \in \tilde{A} : m, n \in NL_{ij} \} \]
\[ KL_{ci} = \{ k : ck \in A_f, \exists n \in NL_{ij}; D_{kn} = 1 \} \]

\( NL_{ij} \) is the set of nodes on the portion of the directed loop from node \( i \) to node \( j \). \( A_{ij} \) is the set of arcs on the directed loop from node \( i \) to \( j \). \( KL_{ci} \) is the set of destinations of cell \( c \) with their \( D \) station on the directed loop from node \( i \) to \( j \). Now suppose \( P_c \) portion of the outflow of cell \( c \) is transferred via node \( i \), and the remaining, \( P_{cj} \), via node \( j \). The total contribution of cell \( c \) in the outflow \( \times \) distance is

\[ O_c = \sum_{k \in KL_{ci}, mn \in A_{ij}} t_{ckmn} l_{mn} + L_{ij}(f_c P_{ci} - \sum_{k \in KL_{ci}, mn \in A_{ij}} f_{ck}) + \sum_{k \in KL_{cj}, mn \in A_{ij}} t_{ckmn} l_{mn} + L_{ji}(f_c P_{cj} - \sum_{k \in KL_{cj}, mn \in A_{ij}} f_{ck}) \]

The above expression is minimized when

\[ f_c P_{ci} = \sum_{k \in KL_{ci}} f_{ck} \Rightarrow P_{ci} = \sum_{k \in KL_{ci}} f_{ck} \]
\[ f_c P_{cj} = \sum_{k \in KL_{cj}} f_{ck} \Rightarrow P_{cj} = \sum_{k \in KL_{cj}} f_{ck} \]

and the total contribution of the outflow of cell \( c \) is

\[ O_c = \sum_{k \in KL_{ci}, mn \in A_{ij}} t_{ckmn} l_{mn} + \sum_{k \in KL_{cj}, mn \in A_{ij}} t_{ckmn} l_{mn} \]

The proof is extendible to cells with more than two non-adjacent edges alongside the loop.

### 4.3.2 LP-Relaxation of D Stations

Although decision variables corresponding to the delivery station locations are left as real variables, they automatically come out integer.

**Theorem 5** Given cell \( c \) with no disconnected edge alongside the loop, the LP-relaxation of the problem coincides with the IP solutions with respect to \( D_{cn} \) variables.
Proof: Suppose node $i$ is selected as a $D$ station of cell $k$. The total contribution of the inflow of cell $k$ in the objective function is $I_k^i$. If cell $k$ splits its inflow among its candidate $D$ stations, its contribution in the objective function is equal to

$$I_k^i = \sum_{i \in \mathcal{N}_k} D_{ki} I_k^i,$$

Suppose

$$I_k^* = \text{Min} \{ I_k^i \}$$

then

$$I_k = \sum_{i \in \mathcal{N}_k} D_{ki} [(I_k^i - I_k^* + I_k^*)]$$

or

$$I_k = I_k^* + \sum_{i \in \mathcal{N}_k, i \neq i^*} D_{ki} (I_k^i - I_k^*)$$

The above expression is minimum when

$$D_{ki} = 0 \quad \forall i \neq i^*$$

Therefore, each destination cell in order to minimize its contribution in the objective function will end up with only one $D$ station. The only exception is when there is a node $i \neq i^*$ where $I_k^i = I_k^*$. 

**Corollary 4** If cell $k$ has two nodes $i, j$ on its boundary such that

$$\sum_{c \in CL_{ki}} f_{ck} / L_{ij} = \sum_{c \in A_f} f_{ck} / (L_{ij} + L_{ji})$$

(4.24)

$$CL_{ki} = \{ c : c \in A_f, \exists n \in NL_{ij}, P_{cn} = 1 \}$$

then there are more than one node satisfying $I_k^i = I_k^*$. That is, the LP values for $D_{ki}$ and $D_{kj}$ variables are not necessarily integer. $CL_{ki}$ is the set of cells having a non-zero flow with cell $k$, and their pick-up station located on the path $ij$ on the directed loop.
Proof: If node $i$ is assigned as the delivery station of cell $k$, the inflow contribution of this cell in the objective function is

$$I_k^i = \sum_{c \in CL_{k,j}} \sum_{mn \in A_{ij}} t_{ckmn}l_{mn} + \sum_{c \in CL_{k,j}} \sum_{mn \in A_{ij}} t_{ckmn}l_{mn} + \left( \sum_{c \in CL_{k,j}} f_{ck}(\sum_{mn \in A_{ij}} l_{mn}) \right)$$

Similarly, when node $j$ is selected as $D$ station of cell $k$, its contribution in the total flow is

$$I_k^j = \sum_{c \in CL_{k,j}} \sum_{mn \in A_{ij}} t_{ckmn}l_{mn} + \sum_{c \in CL_{k,j}} \sum_{mn \in A_{ij}} t_{ckmn}l_{mn} + \left( \sum_{c \in CL_{k,j}} f_{ck}(\sum_{mn \in A_{ij}} l_{mn}) \right)$$

The model is indifferent between the nodes $i, j$ if $I_k^i = I_k^j = I_k^j$. Since sum of the two first terms in $I_k^i$ and $I_k^j$ are equal, their third terms have to be equal

$$\left( \sum_{c \in CL_{k,j}} f_{ck}(\sum_{mn \in A_{ij}} l_{mn}) \right) = \left( \sum_{c \in CL_{k,j}} f_{ck}(\sum_{mn \in A_{ij}} l_{mn}) \right)$$

and in a straight forward manipulation

$$\sum_{c \in CL_{k,j}} f_{ck}/L_{ij} = \sum_{c \in CL_{k,j}} f_{ck}/(L_{ij} + L_{ji})$$

In words, the above equation states that the total inflow of cell $k$ divided by the length of the loop is equal to its inflow from the cells with their $P$ stations between $i$ and $j$ divided by the length of the directed loop in this interval. It is an extremely rare condition. In our example, cell D ended up with non-adjacent edges along the optimal loop. However since the above proportionality is not satisfied, its Delivery station variables came out integer in LP-relaxation.

Based on the justifications stated in this section, the majority of P and D variables are removed from the set of integers. They automatically come out integer.

**Corollary 5** Given cell $c$ with only one outflow which is to its adjacent cell $k$. If cell $k$ has more than one inflow, then $P_{cn}$ variables automatically come out integer.

Proof: As stated in theorem 4, cell $c$ may split its outflow among its common nodes with cell $k$. However, based on theorem 5 the destination cell ends up with only one
Therefore, the optimal values for P variables of the origin cell end up integer.

Based on the above theorem, all D variables are removed from the set of integers. In the 11-cell example, only 15 out of 86 P and D station location variables require branching. By not more than doubling the number of commodities, most station variables come out integer in the LP-relaxation.

As corollaries 3-4 state, few integer variables while are expected to come out 0 or 1 in the LP-relaxation, may end up fractional. To avoid having few integer variables with fractional values, these variables are retained among the integers, but are assigned the last priorities for branching.

### 4.3.3 Appropriate Values for M

Constraints 4.14 and 4.15 state a relationship in which a variable could get a non-zero value only when another variable is greater than zero. Big M is used to formulate these dependencies. There are two difficulties with very large values for M.

1) Multiplying a small number by a large M, could invalidate the formulation.

2) A larger M, results in a larger gap between of LP-relaxation and IP solutions.

Recall the set of binary variables \( Y_{mn} \), the variable is equal to 1 if edge \( mn \) is on the loop, and 0 otherwise. A large M in constraint 4.14, guarantees that if an arc is on the final path, then there is no restriction on the flow passing it. However, if \( Y_{mn} \) is very small and M is very large, their multiplication may result in a number greater than \( \sum t_{cmn} \). Therefore, the model may end up with some flows passing through both directions of an edge, even both directions of an edge which is not on the loop.

An upper bound for the flow passing through an arc is \( \sum_{c} f_{c} \), or 485 in the prototype example. Therefore constraint 4.14 is replaced by

\[
\sum_{c \in C} t_{cmn} \leq (\sum_{c \in C} f_{c}) X_{mn} \quad \forall mn \in A
\]

The feasible region of the LP-relaxation gets closer to integer solutions if M gets smaller.
The value of $Y_{mn}$ in the LP-relaxation is

$$Y_{mn} = X_{mn} + X_{nm} = \sum_{c \in C} (t_{cmn} + t_{cfn})/M$$

The bigger the $M$, the smaller $Y_{mn}$. Therefore, Branch sets $Y_{mn} = 0$, while a strictly positive flow has passed through the edge $mn$. Bound foresees an increase of $l_{mn}(t_{cmn} + t_{cfn})$ in the objective function. That is indeed the increase a bi-directional conventional configuration having nothing in common with a uni-directional single loop. A wrong prediction in Bounding, followed by a wrong selection in Branching, could turn the problem into unsolvables.

The $M$ value of constraint 4.14 could still get smaller at the expense of increasing the number of constraints, i.e., replacing the constraint by the following

$$t_{cmn} \leq f_c X_{mn} \quad \forall mn \in \bar{A} \quad \forall c \in C$$

To retain validity of the formulation, and to have a stronger LP-relaxation, the smallest feasible value of $M$ is identified for each specific constraint.

Following the same line of reasoning, the smallest feasible $M$ for constraint 4.15 is equal to the number of cells adjacent to the node. By increasing the number of constraints, the final form of this constraint is written as

$$P_{cn} \leq 0.5 \sum_{mn \in \mathcal{Y}_n} Y_{mn} \quad \forall c \in C_n \quad \forall n \in N$$

$$D_{cn} \leq 0.5 \sum_{mn \in \mathcal{Y}_n} Y_{mn} \quad \forall c \in C_n \quad \forall n \in N$$

### 4.3.4 LP-Relaxation of Edges

There are 3 alternatives for defining commodities.

1) Define the outflow of each row of the FT-chart as a commodity. Each flow has one origin and one or more destinations. This alternative requires $|C| \times |N|$ sets of multi-commodity flow balance constraints. The nature of the LP-relaxation of this alternative was discussed in Sections 4.3.1, 4.3.2, 4.3.3. The LP-values of a sub-set of P station variables are not integer.
2) Define the outflow of each row as well as the inflow of each column of the FT-chart each as a commodity. A new set of flows corresponding to the columns of the FT-chart (except columns with only one element) are added to alternative 1. Each new flow has one destination and more than one origin. A tight upper bound for the number of multi-commodity flow balance constraints in this alternative is $2|C| \times |N|$. This modification enforces all $P$ station decision variables to come out integer in the LP-relaxation. The proof is similar to that of integrally of $D$ stations in theorem 2. Therefore, $D_{cn}$ variables by row outflow and $P_{cn}$ variables by the column inflow constraints come out integer in the LP-relaxation.

3) To define each element of the FT-chart as a commodity. Here the number of constraints is $|A_f| \times |N|$. The average value for $A_f$ in AGVS material flow system design literature is around $2|C|$, which implies a $200/|C|$ intensity. This intensity of flows is acceptable for product layouts, while the intensity of flow in process layout is higher. An assumption of $|C|^2/4$, i.e., an intensity of 25 percent, seems reasonable. We will examine our model against $200/|C|$, 25, and 50 percent intensities.

By defining each element of the FT-chart as a commodity, the flow balance constraint 4.18 is replaced by

$$f_{ck}P_{cn} + \sum_{mn \in \tilde{A}} t_{ckmn} = f_{ck}D_{kn} + \sum_{nm \in \tilde{A}} t_{cknm} \quad \forall n \in N \quad \forall ck \in A_f$$

(4.25)

By this modification, not only the value of all station variables come out 0/1 in LP-relaxation, but the $M$ values reduce from $\{f_c : \forall c \in C\}$ to $\{f_{ck} \forall ck \in A_f\}$. Therefore, the flow feasibility constraint, constraint 4.14 is finally replaced by

$$t_{ckmn} \leq f_{ck}X_{mn} \quad \forall mn \in \tilde{A} \quad \forall ck \in A_f$$

(4.26)

The LP values for edge variables get closer to 0 or 1.
4.4 Taxonomy of Variables for Branch and Bound

The fundamental insight gained in the last section is implemented to improve the efficiency of the solution procedure. The branching priority of variables has a substantial impact on the efficiency of the solution procedure. Four main layers of binary variables are defined as; edges, arcs, P stations, and D stations. The highest branching priority is assigned to edges, second to arcs, third to P stations, and last to D stations. At each node of branch and bound, as long as the value of an edge variable is 0, its corresponding arc values are set to 0. If the values of all edges incident to a node are equal to 0, then all station variables corresponding to that node are set to 0.

An alternate hierarchy of variables is 1) nodes, 2) edges, and 3) arcs, 4) P stations, 5) D stations. A node could exist on the loop only if at least one of its adjacent nodes exists. When there is a branch to 0 on a node, all edges incident to it are automatically set to 0. The following constraints are added to improve the quality of node variables in the LP-relaxation.

\[ V_n \leq \sum_{mn \in A} V_m \quad \forall n \in N \]

\[ X_{mn} + X_{nm} \leq V_n \quad \forall mn \in \tilde{A} \quad \forall n \in N \]

\[ V_n \leq 0.5 \sum_{mn \in Y_n} Y_{mn} \]

The binary variable \( V_n \) is equal to 1 if the node is on the loop, and 0 otherwise. The computation times were better under the first taxonomy.

Now consider a formulation where its LP-relaxation after branching on a set of variables, either coincide with or is very close to IP values for another set of variables. When Branch and Bound approaches the second set, most of them are already integer and do not require branching. The remaining are close to their integer value, therefore the solution procedure branches on the right direction. Locating these later variables among those having a lower branching priority is almost like removing them from the
set of binary variables. Compare it with when we first branch on these later variables! An example of this situation is binary variables corresponding to station locations of our model. As we proved, even in alternative 1, after branching on edges, a majority of $P$ station variables end up 0 or 1, and when approaching the next layer, almost all $D$ stations variables are already 0 or 1.

4.4.1 Priorities within Nodes and Edges

Priorities within each layer of nodes and edges are defined using two heuristics. The first heuristic defines the variable and value ordering for nodes. The weight of a node depends on the total flow of its adjacent cells. It shows the relative desirability of having the node on the loop, i.e., to branch on it first. Low priority nodes are not only branched on last, but they are first branched on 0.

**Heuristic 1 (Nodes Weight)**

*Begin*

1-SET weight of all nodes = 0
2-FOR(all origin cells $c$
3-\hspace{1cm}FOR(all destinations $k$ of cell $c$
4-\hspace{2cm}IF($c$ and $k$ are adjacent
5-\hspace{3cm}FOR(their common nodes $m,n$
6-\hspace{4cm}ADD $f_{ck}$ to the weights of nodes $m$ and $n$
7-\hspace{2cm}else
8-\hspace{2cm}FOR(all nodes on cells $c$ and $k$
9-\hspace{3cm}ADD $f_{ck}$ to the weight of the node

*End*
To clarify, regarding outflow of cell $D$ to cell $G$ in the prototype example, 15 units is added to the weights of their common nodes, while 25 units is added to the weight of all nodes on cells $D$ and $M$.

The node priority vector is defined as a permutation of integers $1, \ldots, |N|$ denoted by $P = P_1, \ldots, P_{|N|}$; where $P_i$ is the node with $i$th highest weight. In our example, node 3 with weight of 170 has the highest branching priority.

Ties can be broken based on criteria such as, having an adjacent node with higher weight, having higher sum of weights of adjacent nodes, being on a cell with a smaller number of edges, having smaller number of adjacent nodes. We used the first criterion. The rationale is a node closer to a node on the loop has higher potential to be on the loop. To clarify, nodes 6 and 11 with weight of 120 are tied, node 6 is adjacent to 5 with weight of 150, while 11 is adjacent to 3 with weight of 170, node 11 gets higher priority.

The node priority vector in the prototype example is $3, 5, 15, 11, 6, 12, 16, 7, 8, 2, 18, 9, 4, 14, 13, 10, 17, 1$. As we will see later, the first 4 nodes, and 8 out of the first 10 nodes are on the optimal loop.

The second heuristic translates the priority of nodes into the priority of edges. Edges connecting nodes with higher priorities get higher priority for branching.

**Heuristic 2 (Edges Weights)**

1. **Begin**

2. **0-LET** $PE_{P_iP_j}$ denote the priority of edge formed by nodes $P_i$ and $P_j$.

3. **1-** **SET** Priority = 1, and $S = \{P_1\}$

4. **2-FOR**{$j = 2 \cdots |N|$}

5. **3-** **SET** $S = S \cup P_j$

6. **4-FOR**{$i = 1, \ldots, j - 1$}

7. **5-** **IF**{$nodes P_i$ and $P_j$ form an edge}

8. **6-** $PE_{P_iP_j} = \text{Priority}$
7- \textit{INCREASE Priority by 1.}

})

}

}

\textit{End}

To clarify, nodes 3 and 5 have the first and second priorities, edge 3-5 does not exist. The third node is 15, however, 3-15 and 5-15 do not exist too, the fourth node is 11, edge 3-11 gets the first priority. In order to utilize the capabilities of our IP solver, instead of assigning a unique priority to each edge, we have grouped them into 5 sub-layers. Elements of each sub layer have the same priority. The sub layers are \((3-11,5-6,12-15,11-12,15-16,7-12)\), \((5-8,7-8,2-3,15-18,16-18,6-9)\), \((9-16,8-9,3-4,4-5,4-7,11-14)\), \((13-14,10-11,2-10,10-13,17-18,14-17)\), \((13-17,1-6,1-2,1-4)\). Edges with high priority are not only branched on first, but also first branch on 1. The reverse is true for edges with low priority.

4.5 Computational Considerations for Loaded Vehicle Travel

All computations were carried out using CPLEX 3.0 on a SUN Sparc station Model 20.

The loaded vehicle travel (flow \times distance) objective function is defined as follow

\[
\text{Min } LT = \sum_{c \in C} \sum_{m \in A} l_{mn}t_{cmn}
\]  \hspace{1cm} (4.27)

We are unaware of computation times of [TS2] and [ST3] models for single loop material flow system. However, Sinriech and Tanchoco [1992a] report the CPU times for a fixed loop. Given a fixed loop, their model finds the optimal station locations to minimize loaded vehicle travel. It takes 20-1200 seconds of a GOULD NP 1 using a modified version of CPLEX to find the optimal location of stations in the 11-cell.
example. They identify the number of binary variables as the source of complexity in the problem. As explained in section 4.3, we believe that the gap between the feasible region of the LP-relaxation and IP solutions is the most important factor in the solution time. Indeed, if the loop is fixed, a strong majority of binary variables come out integer in LP-relaxation of our formulation.

When the loop is not fixed, there are 444 valid loops in the 11-cells example. After screening them through a set of 3 rules, 17 loops are left. The station location problem have to be solved for these loops. By combining the number of remaining loops and the solution times to find the optimal locations of P/D stations on a given loop, it seems finding the optimal loop requires 10370 seconds. This time excludes the time required to identify all loop, as well as the time required to screen inferior loops using the set of three rules.

In [ST3] some preprocessing is proposed to improve solution times. However, except for the number of branch and bound nodes for a specific fixed loop on the 11-cell example, no computational results are reported. The number of nodes to find the optimal location of P/D stations along a specific fixed loop is 15. Although not of the same nature, the number of nodes for our model to design both the loop and the location of stations is 14.

The optimal solution of our model for the 11-cell example is shown in Figure 4.3. The solution time was 125 seconds. Since the matrix of coefficients is quite tall and the primal problem is highly degenerate, the dual simplex is a better routine to solve the problem. Applying the dual simplex reduced the solution time to 45 seconds. These times are for the case of assuming the total outflow of each cell as a commodity (Alternative 1). Under this formulation, the value of the objective function at the root node was 43 percent of its optimal integer value. By considering each element of the FT-chart as a commodity (Alternative 3), this value was upgraded to 80 percent of the IP optimal objective function. The solution time reduced to 12 seconds. The optimal solution of Figure 4.3 obviously shows a high potential for
generating a tandem configuration out of the loop, or adding a short cut to it. That is to reduce the blocking difficulties of having more than one vehicle on the loop.

To examine the efficiency of the solution procedure, eight sets of quite dense flows were generated from the uniform distribution of (0,100) for the prototype example. Half of the FT-charts were 25 percent filled with strictly positive flows. The remaining are 50 percent dense, i.e., each cell on the average has 5 outflows and five inflows with other cells. Computational results are summarized in Table 4.2.

The problem was first solved using the primal simplex for both root and subsequent nodes. Average CPU times for 25 and 50 percent dense problems were 2501 and 4795 seconds respectively. As shown, CPU times are quite robust for 50 percent dense problems, a range to mean ratio of 0.1. While it is fluctuating for the other set, the range to mean ratio of CPU times is 1.58. It could be a result of more uniformity in more dense FT-charts. Uniformity of a FT-chart is measured from two view-points;
uniformity of the flows, and uniformity of the distribution of the flow of a cell among its adjacent and non-adjacent cells.

Both network simplex and dual simplex outperformed primal simplex at the root node. In 3 out of 4 of the 25 percent dense problems, the dual performed better than network, while for 50 percent dense problems the reverse was true.

In subsequent nodes, dual dominated primal by very large gaps. This is because the number of constraints is more than the number of variables, and the primal is extremely degenerate.

Use of the steepest edge pricing improves the solution time in all problems.

For back tracking, we examined node selection strategies of: 1) the node with best objective function while remaining variables are left linear, and 2) the node with the best estimated integer objective function when all integer infeasibilities are removed. They both outperformed the depth first search. However the second strategy had better results especially on more dense problems.

Finally the intelligent taxonomy of variables was defined. The Branch and Bound developed on this basis realized ultimate significant reductions in CPU times. The average CPU time for the best procedure implemented on the first and second sets of problems is 65 and 417 seconds respectively.

CPLEX by itself is still another source of fluctuation in CPU times. Its running time on different computers is not proportional to the speed of the computer. This is explained in part by the dependency of CPLEX parameters on the corresponding computer configuration and capabilities. Furthermore, the software is sensitive to the coefficients in the model. To clarify, flow feasibility constraint 4.14, and multi-commodity material flow balance constraints 4.18 were implemented in 3 slightly different forms; 1) as they already are, 2) $t_{ckmn}$ was assumed as a variable in the range of 0 to 1, and was multiplied by $f_{ck}$ in both constraints, and 3) the same as 2 but $f_{ck}$ was removed from the both sides of the both constraints. Fluctuations of up to 60 percent were observed in CPU times.
Table 4.2: Computation times for randomly generated flows using different solution procedures

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>1886</td>
<td>1955</td>
<td>1953</td>
<td>1958</td>
<td>1938</td>
<td>3695</td>
<td>3684</td>
<td>3693</td>
<td>3711</td>
<td>3695</td>
</tr>
<tr>
<td>Non zeros</td>
<td>9315</td>
<td>9655</td>
<td>9655</td>
<td>9643</td>
<td>9567</td>
<td>18095</td>
<td>18059</td>
<td>18103</td>
<td>18167</td>
<td>18106</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CPU TIMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primal</td>
</tr>
<tr>
<td>Dual</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dual simplex on subsequent nodes, steepest edge pricing, and the followings on root node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primal</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>Dual</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dual on subsequent nodes, steepest edge, best integer estimate node selection, and the followings on root node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
</tr>
<tr>
<td>Dual</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dual on root and subsequent nodes, steepest edge, intelligent B and B, and the following Lower Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
</tr>
<tr>
<td>Integer Est.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network on root, Dual on subsequent, steepest edge, and intelligent B and B, and following L.B.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
</tr>
<tr>
<td>Integer Est.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statistics for Best Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time</td>
</tr>
<tr>
<td>LP O.F.</td>
</tr>
<tr>
<td>IP O.F.</td>
</tr>
<tr>
<td># of Nodes</td>
</tr>
<tr>
<td># of Iterations</td>
</tr>
</tbody>
</table>
A loop flow pattern is a flexible configuration. It could simply handle any diversification in products and deviation in the production plan. Whenever there is a change in product type, or production level, the new optimal solution is identified shortly. For the case of 11-cell example, even for very dense F-T charts, the new solution is identified in a few minutes. The time required to implement the new solution depends on the number of uncommon edges between the existing design and the new solution.

### 4.6 Extensions

#### 4.6.1 Multiple stations per cell

P variables are now interpreted as the proportion of the flow of a cell shipped through a specific station. In this case, if a station variable is greater than zero, the station is constructed. Since station variables are removed from the set of integers, the solution times get smaller. By increasing the number of stations the loaded vehicle travel distance is reduced. The value of the objective function for multiple station per cell is a lower bound for the case of one P/D station. However, it does not necessarily reduce loaded vehicle travel time. When an AGV travels from one station to another, it accelerates, reaches a constant velocity, and decelerates. By allowing more stations alongside the loop, the average distance between stations is reduced. The shorter the distance of the travel, the slower the average velocity of the AGV. Furthermore, it could increase blocking as well as construction and space occupancy costs.

Exploring the possibility of having more than one station per cell is suggested for those cells; 1) adjacent to many other cells, and 2) with non-adjacent edges alongside the loop. However, for the first case, instead of increasing the number of AGVs stations, some stationary material handling equipment could be added to transfer material between adjacent cells.

The optimal solution for our example under the assumption of multiple P/D stations per cell is shown in Figure 4.4. There are four new P stations (two for each
of cells D and J) and three new D stations (each for cells D, G, K) added to the previous twenty P/D stations. By constructing these seven new P/D stations, the total loaded distance travel is reduced from 3650 to 850. As a by-product of relaxing stations from being integer, the solution time reduced to 3 seconds. In other words, unlike other models of locating stations on a loop, the solution time of our model decreases when multiple stations per cell are allowed.

4.6.2 The Cost Model

The cost based model integrates loaded vehicle travel, loop, and station costs. Minimize $TC =$

$$
\sum_{c \in \mathcal{A}_i} \sum_{mn \in \mathcal{A}} Q^{t}_{mn} t_{mn} t_{ckmn} + \sum_{mn \in \mathcal{A}} Q^{c}_{mn} t_{mn} X_{mn} + \sum_{c \in \mathcal{C}} \sum_{n \in \mathcal{N}_c} Q^{t}_{cn} (P_{cn} + D_{cn}) \quad (4.28)
$$
$Q_{mn}^t$ is the cost of one unit distance unit loaded travel on arc $mn$. $Q_{mn}^c$ is the annuity of the construction cost, space occupied, maintenance, and other costs associated with edge $mn$. $Q_{cn}^t$ is the annuity of the construction cost, space occupied, maintenance, and other costs associated with station $cn$. The model requires each flow of FT-chart to be assumed as a commodity, i.e., each commodity has one origin and one destination. A set of new binary variables are added to the model. The binary decision variable $p_{ckn}$ is 1 if node $n$ is selected as pick-up station at cell $c$ for the flow $ck$, and 0 otherwise. Similarly, the binary variable $d_{ckn}$ is 1 if node $n$ is selected as delivery station at cell $k$ for flow $ck$, and 0 otherwise. Whenever one or more outflow(inflow) of a cell is Picked-up (Delivered) at a specific node, the corresponding P(D) station is constructed. $P_{cn}$, $D_{cn}$ variables need not be among integer variables. Constraints 4.11, 4.12, as well as 4.15 are removed. The following constraints are added to the problem.

\[
P_{cn} \geq p_{ckn} \quad \forall n \in N_c \quad \forall ck \in A_f \tag{4.29}
\]

\[
D_{cn} \geq d_{ckn} \quad \forall n \in N_c \quad \forall kc \in A_f \tag{4.30}
\]

\[
p_{ckn}, d_{ckn} \in \{0, 1\}
\]

The material flow balance constraint is replaced by the following

\[
f_{ck}p_{ckn} + \sum_{mn \in \mathcal{A}} t_{ckmn} = f_{ck}d_{ckn} + \sum_{nm \in \mathcal{A}} t_{cknm} \quad \forall n \in N \quad \forall ck \in A_f \quad \forall c \in C \tag{4.31}
\]

### 4.6.3 Computational Considerations

To examine the cost integrated model, we assumed 20c as the cost of handling one unit load per unit distance, $(18I_{mn} + 80)$ as the annuity of all initial costs, and all other yearly cost of each arc on the network, and $25 as the annuity of all initial costs and all other yearly costs of each station. The optimal loop which is shown in 4.5 was obtained in 15 seconds. The total cost of opening and constructing each arc is high compared to the loaded vehicle travel costs. The model has not designed all
Figure 4.5: Optimal solution for the cost objective functions
arcs required to minimize the loaded vehicle travel. The cost of constructing each P/D station is relatively low, the model has taken advantage of it and has come out with 25 stations, which is between the minimum and the maximum possible number of stations.

The solution time of this model is not independent of its cost parameters. For example, suppose saving in travel costs resulted from opening an edge or station is close to its construction cost. and the model should examine to trade them off or not, then the CPU times increase.

Figure 4.6 is one way of illustrating such a relationship. In this figure, P/D station and edge costs are fixed. The cost of one unit distance of unit load travel is shown on the horizontal axis. It is gradually increased in each independent run. Therefore in some points, the saving in travel costs approaches its trade-off point with the station cost, or with cost of opening a new edge. These points are shown on the graph. From computation time point of view, it takes more time for the model to decide whether to construct a new station or not. In other words, at the proximity of these points, lower bounds of a larger number of nodes are less than the integer value of the objective function. Therefore, branches are fathomed at much lower nodes of the search tree.

The CPU times corresponding to these situations are those around the local maxima on the left hand side of the curve. The global maximum occurred when cost savings of all entities are close to their trade-off points. Note that the loaded travel cost function does not decrease monotonically by decreasing rate. That is because the unit cost of all edges are not equal. The unit cost is composed of a variable cost proportional to the length of the edge, and a fixed cost. The fixed cost of opening each edge is distributed over the length of the edge. CPU time reduces on the right hand side of the curve. This is due to high travel costs, when branches of not constructing any required edge or station are fathomed at earlier nodes of the search tree.
4.6.4 Bi-directional Single Loop

To allow bi-directional flow, all constraints containing $X_{mn}$ variables are removed. Constraint 4.14 is replaced by the following

$$t_{ckmn} + t_{ckm} \leq f_{ck} Y_{mn} \quad \forall mn \in A \quad \forall ck \in A_f$$

(4.32)

Furthermore, formation of some composite configurations may be allowed in a bi-directional loop. We may allow patterns in which two sub-tours are connected by a single node, Figure 4.7(a). To allow formation of such a pattern, we first eliminate constraint 4.8 on 4-edge intersections, and add the following for not having 3 edges on the flow path at these intersections.

$$\sum_{\substack{\text{in } Y_n \ni mn \in E \to \text{out } \forall n}} (1 - Y_{in}) \geq (1 - Y_{mn}) \quad \forall mn \in A \quad \forall n : |Y_n| = 4$$

(4.33)

$Y_n$ is the set of nondirected edges incident to node $n$. By eliminating constraint 4.8 which was among the original constraints, and constraint 4.33 added above, formation...
of degree 3 nodes are also allowed. Therefore, a tandem configuration may appear, Figure 4.7(b). To accelerate this tendency, no sub-tour elimination constraint is included in this formulation.

Note that the above degree 3 modification is not easily applied on a uni-directional loop. The main reason is the presence of the arc balance constraint. In order to satisfy this constraint for both degree even and degree 3 nodes, an additional integer variable is required. The variable accepts values of 0,1,2 in order to modify the arc balance constraint which is not satisfied any more. Although one may apply a limit on the number of degree 3 nodes, solving the model for uni-directional loop is restricted to very small problems.

The optimal solution for the bi-directional loop is shown in Figure 4.8. The value of the objective function is 2245, the solution time was 34 seconds. Insight derived from analyzing both uni-directional and bi-directional optimal solutions could be used to generate a tandem configuration or a segmented flow topology out of the bi-directional loop. That is mainly to reduce the difficulties of having more than one vehicle in any zone on the loop.
Figure 4.8: Optimal solutions for bi-directional loop
Conventional Configuration

By replacing constraints 4.8 and 4.9 with the following, the model is extended to a conventional configuration.

\[
M \sum_{mn \in \mathcal{A}} X_{mn} \geq \sum_{nm \in \mathcal{A}} X_{nm} \quad \forall n \in N \tag{4.34}
\]

\[
M \sum_{nm \in \mathcal{A}} X_{nm} \geq \sum_{mn \in \mathcal{A}} X_{mn} \quad \forall n \in N \tag{4.35}
\]

M is replaced by 2 and 3 for degree 3 and 4 nodes respectively. The above pair of constraints ensure that if a node has one incoming (outgoing) arc, it has at least one outgoing (incoming) arc. In almost all models developed for conventional configuration, the station locations are fixed. The objective of the model is to assign direction to the arcs in order to minimize the total loaded vehicle travel. Except Goetz and Egbelu [1990], we are unaware of any other formulation for a conventional configuration in which the direction of arcs and the location of P/D stations are designed simultaneously. Our formulation differs from theirs in the following aspects;

1) Shortest routes for every combination of P and D stations are among the inputs to their model. The difficulty is not only to input the shortest routes, but also due to a considerable number of shortest routes.

2) Summation of multiplication of groups of four binary variables appear in their objective function. Using Lawler's substitution, they linearize the multiplication of a set of variables by a 9-subscript linear variable, and regulate the relationship between the linear and binary and variables. We are unaware why they use a 9-subscript instead of 8-subscript variable, and why they locate it among the integers. They develop some rules to reduce the huge number of constraints.

3) For the purpose of strong connectivity, they have used constraints of at least one incoming and one outgoing arc for all nodes and all combinations of nodes. Therefore, all arcs are constructed in any feasible and optimal solution. We have implemented a modification of DFJ constraints along with a closed dummy flow loop passing all
There are no computational times reported in Goetz and Egbleu [1990]. A large number of binary variables and no attempt to tighten the LP-relaxation leaves the model solvable for only very small problems.

The conventional configuration for the prototype example is shown in Figure 4.9. The value of the objective function is 2060, the solution time was 13 seconds. Having the optimal objective function for the loop configuration to be 3650. We may state that the trade off ratio between the loop and the conventional configuration is $3650/2060 = 1.77$. Furthermore, we examined this ratio for 8 layouts of size ten with their corresponding FT chart. The overall ratio of loaded vehicle travel distance in a conventional configuration to that of a single loop was 1.98. The standard deviation was 0.24. Note that on the other hand, capital cost of a conventional configuration is higher, and its dispatching, vehicle routing, and traffic management is more complicated.
4.7 Conclusion

In this chapter, we developed an exact formulation for the integrated design of the flow pattern and the location of Pick-up and Delivery stations. Computational results and potential extensions of the model seem promising.
Chapter 5

The Optimal Loop Flow System for Empty and Loaded AGV Travel

In this chapter we develop an exact formulation to design a loop material flow system for Automated Guided Vehicles. The objective of the model is to minimize the total empty and loaded vehicle travel distance. Edges of a fixed block layout are candidates for the flow path. The block layout is augmented with a set of new nodes at the mid-points of edges as the candidate locations for Pick-up and Delivery stations. The situation in which original nodes of the block layout are candidate for P/D station locations is also examined. Three environments of Fixed Loop/Variable Station, Variable Loop/Fixed Station, Variable Loop/Variable Station are modeled. The emphasize is on the model in which the loop and the locations of stations are variable and one P and D station is allowed per cell.

5.1 Introduction and Review

Sinriech and Tanchoco [1992] conduct a simulation study to compare throughputs of a single loop and a conventional configuration. Different empty vehicle dispatching rules are implemented. Regardless the dispatching rule, the throughput of the single loop is much robust than that of the conventional configuration. Their conclusion is
that the impact of the empty vehicle travel on the performance of a single loop is negligible.

The loop implemented in this comparison is not the optimal solution of their loaded vehicle travel model. The conventional configuration is also not an optimal solution for loaded vehicle travel. The optimal loop and conventional configuration for loaded vehicle travel of their example was obtained using the model developed in chapter 4. In this chapter, it is shown that ignoring empty vehicle travel has a substantial impact on the optimal solution.

We develop an integrated model to design a loop, determine its direction and location of P/D stations in order to minimize total loaded and empty vehicle travel distance. The 11-cell example of Sinriech and Tanchoco [1993] is used to explain our formulation, solution procedure, and computational considerations. The layout and its From-To chart are shown in Figure 5.1 and Table 4.1. Edges of the layout are candidates for the segments of the loop. A set of degree two nodes inserted somewhere along each edge, shown by circles, are candidate locations for P/D stations. In all models developed to design a single loop, the P/D stations are located on the intersection nodes, and the objective is to minimize the total loaded vehicle travel distance. In order to show the impact of the empty vehicles on the optimal solution of these models, we also solve the problem for the case where no degree 2 node is added to the block layout. However, for not ending up with an unrealistic optimal solution in which many P/D stations are clustered together, and to avoid blocking, the configuration of stations on edges is preferred. Note that, in a Mini-Sum problem the intersections are optimal locations for the stations, while this is not the case for Mini-Max problem. We show that the model of this chapter is of Mini-Max nature. All computations are carried out using the general purpose MIP solver CPLEX3.0 on a SUN Sparc station Model 20.

\(^1\)Note that the original degree 2 nodes at corner points of the block layout are not considered as nodes. The pair of edges connected to them are assumed as one edge. We call degree 3-4 nodes intersections, while degree 2 nodes are junctions.
Figure 5.1: The layout of the example from Tanchoco and Sinriech [1992]

<table>
<thead>
<tr>
<th>Environment</th>
<th>Flow Loop</th>
<th>Station Location</th>
<th>Number of Stations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fixed</td>
<td>Variable</td>
<td>One</td>
</tr>
<tr>
<td>2</td>
<td>Variable</td>
<td>Fixed</td>
<td>One</td>
</tr>
<tr>
<td>3</td>
<td>Variable</td>
<td>Variable</td>
<td>One</td>
</tr>
</tbody>
</table>

Table 5.1: Combinations of the assumptions in this chapter

To the best of our knowledge, there is no optimal or heuristic procedure considering the substantial role of the empty vehicle travel on a single loop material flow system. Decomposition and Brute Force search solution procedures are discussed.

5.2 Formal Statement

Let $G_b(N, A)$ be the graph associated with the block layout. $N$ is the set of intersections on the boundaries of the cells as well as the degree 2 nodes inserted on edges connecting intersections. $A$ is the set of nondirected edges. For any pair of adjacent nodes, there exist an edge in $A$. Each face $c \in C$ of the graph represents a cell in
the block layout. In the directed graph \( G_b(N, \tilde{A}) \), each unordered edge \( mn \in A \) is replaced by two ordered arcs \( mn, nm \in \tilde{A} \). Given \( \mathcal{P}_c \) and \( \mathcal{D}_c \) as the set of degree 2 nodes being the candidate locations for pick-up and delivery stations of cell \( c \), one \( \mathcal{P} \) and one \( \mathcal{D} \) station have to be fixed for each cell. A Feasible Loop is a set of directed arcs on a cycle \( \Gamma \) covering the selected sub-set of stations.

There is a (usualy non-planar) graph \( G_f(C, A_f) \) representing the material flow in the block layout. For each pair of cells \( c, k \) with a strictly positive flow from \( c \) to \( k \), there is a directed arc in \( A_f \) connecting \( c \) to \( k \). A function \( f: A_f \to \mathbb{R}^+ \), assigns \( f_{ck} \) as the intensity of material flow to the arc \( ck \in A_f \). Given \( \omega_{mn} \) as the total flow passed through arc \( mn \), and \( \Lambda_{ck} \subseteq \Gamma \) as the set of arcs on the loop from \( \mathcal{P} \) station of cell \( c \) to \( \mathcal{D} \) station of cell \( k \), \( \omega_{mn} \to \omega_{mn} + f_{ck} \quad \forall mn \in \Lambda_{ck} \quad \forall ck \in A_f \). Given \( \Theta_{\Gamma} \) as the maximum flow passed through an arc of the cycle \( \Gamma \), i.e., \( \Theta_{\Gamma} \geq \omega_{mn} \quad \forall mn \in \Gamma \), the optimal cycle, \( \Gamma^* \), is the cycle for which \( \Theta_{\Gamma^*} \leq \Theta_{\Gamma} \quad \forall \Gamma \).

### 5.3 The Optimization Model

#### 5.3.1 Formulation

\[
\sum_{n \in \mathcal{P}_c} P_{cn} = 1 \quad \forall c \in C \tag{5.1}
\]

\[
\sum_{n \in \mathcal{D}_c} D_{cn} = 1 \quad \forall c \in C \tag{5.2}
\]

\[
P_{cn} \leq \sum_{mn \in \tilde{A}} X_{mn} \quad \forall c \in C \quad \forall n \in \mathcal{P}_c \tag{5.3}
\]

\[
D_{cn} \leq \sum_{mn \in \tilde{A}} X_{mn} \quad \forall c \in C \quad \forall n \in \mathcal{D}_c \tag{5.4}
\]

\[
X_{mn} + X_{nm} = Y_{mn} \quad \forall mn \in A \tag{5.5}
\]

\[
\sum_{mn \in \tilde{A}} X_{mn} \leq 1 \quad \forall n \in N \tag{5.6}
\]

\[
\sum_{mn \in \tilde{A}} X_{mn} = \sum_{nm \in \tilde{A}} X_{nm} \quad \forall n \in N \tag{5.7}
\]

\[
\sum_{m \in \mathcal{R}_s, n \in \mathcal{R}_s} (X_{mn} + X_{nm}) \geq 2 \quad \forall s \in S \tag{5.8}
\]
A Loop Material Flow System Design for Manufacturing Plants

\[ t_{ckmn} \leq f_{ck} X_{mn} \quad \forall mn \in \tilde{A} \quad \forall ck \in A_f \]  
(5.9)

\[ f_{ck} P_{cn} + \sum_{mn \in \tilde{A}} t_{ckmn} = f_{ck} D_{kn} + \sum_{mn \in \tilde{A}} t_{ckmn} \quad \forall n \in N \quad \forall ck \in A_f \]  
(5.10)

\[ \sum_{ck \in A_f} t_{ckmn} \leq \Theta \quad \forall mn \in \tilde{A} \]  
(5.11)

Minimize \[ \sum_{mn \in \tilde{A}} l_{mn} X_{mn} \]  
(5.12)

Minimize \[ \sum_{ck \in A_f, mn \in \tilde{A}} l_{mn} t_{ckmn} \]  
(5.13)

Minimize \[ \Theta \sum_{mn \in \tilde{A}} l_{mn} X_{mn} \]  
(5.15)

\[ \forall mn \in A \quad Y_{mn} = \begin{cases} 1 & \text{if the edge } mn \text{ is on the loop} \\ 0 & \text{Otherwise} \end{cases} \]  

\[ \forall mn \in \tilde{A} \quad X_{mn} = \begin{cases} 1 & \text{if the arc } mn \text{ is on the loop} \\ 0 & \text{Otherwise} \end{cases} \]  

\[ \forall c \in C \quad \forall n \in P_c \quad P_{cn} = \begin{cases} 1 & \text{if node } n \text{ is selected as } P \text{ station of cell } c. \\ 0 & \text{Otherwise} \end{cases} \]  

\[ \forall c \in C \quad \forall n \in D_c \quad D_{cn} = \begin{cases} 1 & \text{if node } n \text{ is selected as } D \text{ station of cell } c. \\ 0 & \text{Otherwise} \end{cases} \]  

5.3.2 Terminology

- \( l_{mn} \): Length of arc \( mn \)
- \( f_{ck} \): Intensity of flow from cell \( c \) to cell \( k \)
- \( C \): Set of cells
- \( N \): Set of nodes
- \( A \): Set of undirected edges.
- \( \tilde{A} \): Set of directed arcs.
- \( A_f \): Set of ordered pair of cells \( ck \) with a flow intensity of \( f_{ck} > 0 \)
- \( X_c \): Set of arcs on cell \( c \)
- \( Y_n \): Set of edges incident to node \( n \)
- \( P_c \): Set of candidate nodes for \( P \) station location on cell \( c \)
- \( D_c \): Set of candidate nodes for \( D \) station location on cell \( c \)
- \( N_c \): Set of intersection nodes on cell \( c \)
- \( C_n \): Set of cells adjacent to node \( n \)
\[ S \] : Set of primitive sub-tours
\[ R_s \] : Set of nodes on the cells forming a primitive sub-tour \( s \)
\[ \overline{R}_s \] : Complement of \( R \) with respect to \( N \)
\[ Y_{mn} \] : Binary decision variable corresponding to undirected arc \( mn \), where \( n > m \)
\[ X_{mn} \] : Binary decision variable corresponding to directed arc \( mn \)
\[ P_{cn} \] : Binary decision variable corresponding to node \( n \) as pick-up station of cell \( c \)
\[ D_{cn} \] : Binary decision variable corresponding to node \( n \) as delivery station of cell \( c \)
\[ t_{ckmn} \] : Intensity of flow from cell \( c \) to cell \( k \) passing arc \( mn \)
\[ \omega_{mn} \] : Total flow passed through arc \( mn \)
\[ \Lambda_{ck} \] : The set of arcs on the loop from \( P \) station of cell \( c \) to \( D \) station of cell \( k \)
\[ \Theta_{\Gamma} \] : The maximum flow passed through an arc of cycle \( \Gamma \)

5.3.3 Description

Integer Decision Variables

A binary variable \( Y_{mn} \), where \( n > m \) is defined for each undirected edge \( mn \). A binary variable \( X_{mn} \), where \( m > n \), is defined for arcs \( mn \) on a cell with the smallest number of edges. All other \( X_{mn} \) variables are left real, they automatically end up 0 or 1, Asef-Vaziri and Sriskandarajah [1996]. A binary variable \( P_{cn} \), is defined for each candidate node \( n \in P_c \), \( \forall c \in C \). A binary variable \( D_{cn} \), is also defined for each candidate node \( n \in D_c \), \( \forall c \in C \).

Station Locations

Each cell has one pick-up and one delivery station, constraints 5.1 and 5.2. If more than one P and D stations are allowed for each cell, then the equality sign is replaced by the greater or equal sign. Stations locations of each cell are restricted to the nodes on the boundary of the cell. Stations are located along the flow path, constraints 5.3 and 5.4. These constraints are only required for common nodes of adjacent cells having no flow except with themselves. The constraint is automatically satisfied for other nodes and cells as a result of constraint 5.9 coupled with 5.10.

In the models developed to find the optimal loop for loaded vehicle travel, Tan-
choco and Sinriech [1992], Sinriech and Tanchoco [1993], Asef-Vaziri and Sriskandarajah [1996], P/D stations are located on the intersections of the block layout. We also solve the model for the case where stations are located on the intersections. That is to show how empty vehicle considerations lead to a solution quite different from the optimal solution for loaded vehicle travel.

In this case both $P_c$ and $D_c$ are replaced by $N_c$ in constraints 5.1 and 5.2. In almost all material flow system designs, it is required that the flow pattern covers at least one edge of each cell. Where stations are located on the inserted degree 2 nodes, covering a candidate station location is equivalent to covering an edge of the corresponding cell. For the case of stations on intersections, an additional constraint is required to enforce the loop to cover at least one edge of each cell. Therefore the following constraint is required.

$$\sum_{mn \in N_c} X_{mn} \geq 1 \quad \forall c \in C$$  \hspace{1cm} (5.16)

**Loop Configuration**

The loop is uni-directional, constraint 5.5, there is at most one incoming arc to each node, constraint 5.6. The number of incoming and outgoing arcs at each node are equal, 5.7.

**Sub-Loop Elimination**

The Dantzig, Fulkerson, Johnson [DFJ, 1954] sub-tour elimination constraint is implemented in our model. Our definition of sub-tour differs slightly from its original concept in the traveling salesman problem. Any sub-set of adjacent cells is a potential sub-loop, unless the boundary of the composite shape covers at least one edge of every cell. It is *Primitive* if its own elimination constraint is requires. Constraint 5.8 is sub-tour elimination constraint, where $S$ is the set of *primitive* sub-loops, $R_s = \{ n : n \in NC_c, c \in s \}$, and $\overline{R_s} = N \setminus R_s$. 
Asef-Vaziri *et al.* [1996], showed the following properties regarding sub-tour elimination in a block layout.

1-Since the dual of $G_b(N, A)$, commonly known as the adjacency graph $G(C, A)$, is far from complete, the number of potential sub-tours is less than that of the Traveling Salesman Problem.

2-The collection of adjacent cells where one of them does not have an edge on the boundary of the composite shape is an impossible sub-tour. Furthermore, some sub-tours, while their size is smaller than $\lfloor|C|/2\rfloor$ are complete tour.

3-Unlike the TSP, the sum of the number of cells in a sub-tour $s$, and that of its complement $\overline{s}$ is neither $|C|$ nor constant, $|s| + |\overline{s}| < |C|$. Therefore, the steady state criteria of $|s| \leq \lfloor|C|/2\rfloor$ for a sub-tour to be Primitive, i.e., its own elimination constraint to be included in the model, is replaced by the transient criteria of $|s| \leq |\overline{s}|$.

4-The sub-graph $\overline{s}$ is not necessarily connected. Therefore, if $r$ is the smallest component of $\overline{s}$, the elimination constraint of the sub-tour $s$ is required, if and only if $|s| \leq |r|$.

5-Given the set of primitive sub-tours $S = \{s : |s| \leq |r|\}$. There is a dominant sub-set $S' \subseteq S$, such that given any sub-tour $s \in S \setminus S'$, there exist a sub-tour $s' \in S'$ such that the elimination constraint of $s'$ is at least as strong as that of $s$. Therefore, all primitive sub-tours $s \in S \setminus S'$ are removed from $S$, and it shrank to $S'$.

Furthermore, when material flow is added to the problem, if the flow graph is strongly connected, then a single loop emerges even in the absence of sub-tour elimination constraints. To examine the strong connectivity of the flow graph, a length of 0 is assigned to any edge $ck \in A_f$. For each pair of cells $c, k$, where $ck \in A$ an edge of length of 1 is defined. If there exist a solution to the asymmetric TSP defined on the flow graph, then the graph is strongly connected. Otherwise, for those edges $ck$ having a value of 1 in a feasible or in the optimal solution, a very small flow is added to the original flow graph to make it connected.

The above check and modifications guarantee that without requiring sub-tour
elimination constraints, a single loop is formed. Then the role of DFJ constraint is nothing but tightening the LP-Relaxation. However, even without the above modifications, one may leave the model without sub-tour elimination constraints. If the final flow system is not connected, it may motivate a tandem type configuration (a set of loops). However, these possibilities are not discussed in the present thesis.

Material Flow

Each element \( ck \) in Table 4.1 represents a specific set of parts differentiated from all others. This portion of the model is a multi-commodity flow problem in a loop flow pattern.

The Flow Feasibility Constraint 5.9 states that material can flow only along the loop. If an edge or a node is not allowed on the flow system, its corresponding variable is set to zero. Constraint 5.10 is the multi-commodity material flow balance. Material movement within the cells is not included in the flow constraints. It is assumed that the distances from candidate station locations on the boundary of each cell to the activity center inside the cell are equal.

Elements of the FT chart are expressed in unit loads, since each vehicle carries one unit load at a time. Therefore the \( t_{ckmn} \) variables have to be integer. Fortunately, unlike the general form of the multi-commodity flow problem, there is only one path between every pair of points in a single loop. Therefore, the LP-relaxation of \( t_{ckmn} \) variables is on their IP solutions.

Constraint 5.11 is employed to linearize the nonlinear function of

\[
\text{Min } \{ \text{Max } \{ \sum_{ck \in A_f} t_{ckmn} \} \forall mn \in \bar{A} \}
\]

The minimization of the length of the loop 5.12, loaded vehicle travel distance minimization 5.13, minimization of the maximum flow passing through an arc of the loop, 5.14, and the total loaded and empty travel distance minimization 5.15 are potential objective functions for the proposed configuration.
5.3.4 Empty Vehicle Travel

The followings four approaches are proposed to incorporate the empty vehicle travel into the model.

1) Each destination cell \( k \) in the FT chart, is an origin cell in the empty vehicle FT chart with the total outflow \( f''_k = \sum_{ck \in A} f_{ck} \).

Sending the vehicles back to their origin of their trip, like Venkataramanan and Wilson [1991], or to a central terminal like Majety and Wang [1995] are not appropriate in a single loop configuration. A MIP model could be formulated in which empty vehicles generated at a cell \( c \) are attracted by all cells \( k \in C \) with the attraction factor of total outflow of the cell divided by the total flow in the system. If the empty vehicles assigned by the MIP model from cell \( c \) to cell \( k \) is \( f''_{ck} \), then a new FT chart is prepared where, \( f_{ck} \rightarrow f_{ck} + f''_{ck} \). Goetschals and Palliyil [1994].

2) The above flow graph does not account for all empty vehicle travel. Note that flows added to the initial flow graph are generated at pick-up stations, while empty vehicle flows start from delivery stations. The flow from D to P stations are not included in the above model. To resolve this shortcoming, the FT-chart is returned to its original form. Therefore, there is a set of loaded flow \( f_{ck} \), and a separate set of empty flow \( f''_{ck} \). The origin of \( f''_{ck} \) is the D station of cell \( c \), and its destination is the P station of cell \( k \). In other words, in the material flow balance constraint, regarding \( f_{ck} \), P station is the origin and D station is the destination. While the reverse is true for flow \( f''_{ck} \).

3) Like alternative 2, but all flows \( f''_{ck} \forall c, k \) are assumed to be a single commodity.

4) The analytical measure of the total empty and loaded vehicle travel is the product of the length of the loop and the number of times the loop is traversed. The FT chart developed by the above MIP may be implemented as the flow of the single commodity, of the total empty and loaded vehicle travel is the product of the length of the loop and the number of times the loop is traversed. Since AGVs are of the unit load type, the number of times the loop is traversed is equal to the maximum
flow passed through an edge of the loop. This last approach is the most appropriate indicator of the empty and loaded vehicle travel. However, this objective function has a difficult mathematical form; multiplication of two integer variables which are not restricted to any specific short interval.

5.3.5 Additional Constraints

To have a stronger LP-Relaxation, the following constraint is added to the problem.

\[
\sum_{j \in N \setminus \{n\}} X_{jn} + X_{nj} \geq X_{mn} + X_{nm} \quad \forall m,n \in N \quad (5.17)
\]

It is to lift the LP-relaxation of the "at most degree 2" constraint, and "arc balance" constraint. Given the size of the problems we are concerned, having a stronger LP-relaxation is not very important in designing a loop pattern. It becomes of crucial importance when material flow is added to the problem.

5.3.6 An Upper Bound

The loaded vehicle travel optimization was modeled by Tanchoco and Sinriech [1992], Sinriech and Tanchoco [1993], Asef-Vaziri and Sriskandarajah [1996]. In all formulations, intersections on the boundary of cells are candidate locations for P/D stations. One P and one D station are allowed per cell. The optimal loaded vehicle travel for the prototype 11-cell example of Tanchoco and Sinriech [1992] and Sinriech and Tanchoco [1993] is 3650. This is the sum of the length of each arc times the flow passing through arc. Since there are usually multiple optimal solutions, we conducted two improving post optimization runs. This was to upgrade the loaded vehicle optimal solution to its best design for both loaded and empty vehicle travel.

First, the loaded vehicle objective function set to its optimal value of \( Z_{\text{Loaded}} \) = 3650 was added to the constraints. A new objective function was defined to minimize the maximum flow passing through the loop. Its optimal solution for the maximum flow passing through an arc of the loop is \( Z_F \mid Z_{\text{Loaded}} = 3650 = 95 \). Second, the new
The first upper bound and the optimal solution for loaded/empty vehicle in the 11-cell example

The model was solved for the shortest loop objective function, the optimal value was $Z_L^* = 118$. As stated earlier, the logical measure of effectiveness in evaluating both loaded and empty vehicle travel is the number of cycles the loop is traversed times the length of the loop. Therefore, the optimal solution for loaded vehicle travel, when improving it to its best for both loaded and empty travel is 11210 loaded/empty vehicle travel distance. The solution is shown in Figure 5.2.

The optimization model developed in section 3-1, under the assumption of stations on intersections was solved for MiniMax flow as the primary and length of the loop as the secondary objective functions. $Z_F^* = 65$, $Z_L^*|Z_F^* = 65 = 118$, and the total loaded and empty travel is 7670. This solution is shown in Figure 5.2. Furthermore, for length as the primary and Max flow as the secondary objective function, $Z_L^* = 100$, $Z_F^*|Z_L^* = 100 = 80$, and the total loaded and empty travel is 8000. This solution is
Figure 5.3: The second upper bound for loaded/empty vehicle in the 11-cell example shown in Figure 5.3. The upper bound for loaded and empty vehicle travel is 7670. While under upgraded optimal solution for the loaded vehicle travel is 11210, Figure 4.3. This is at least 46 percent deviation from the optimal solution for loaded/empty vehicle travel.

5.4 Solution

5.4.1 Decomposition

To find the optimal solution for loaded/empty travel, a decomposition and brute force procedure is proposed. In order to solve the master problem with the nonlinear objective function of Length $\times \text{Max}\{\text{Flow}\}$, we decompose it into Loop and Flow problems. Indeed, a set of sub-problems of; Loop, Flow, Flow|Loop, Loop|Flow, Range of Covering Arcs [RCA] are defined in this section.

The Loop Problem is to find the shortest loop with its length in a specific range.
This problem is solved quickly for practical size layouts.

**Loop Problem**

\[ \text{Min} Z_L = \sum_{mn \in \mathcal{A}} l_{mn} X_{mn} \]

s.t. \hspace{1cm} (a) Loop Configuration

\[ (b) \ L_{Lower} + \varepsilon \leq \sum_{mn \in \mathcal{A}} l_{mn} X_{mn} \leq L_{Upper} - \varepsilon \]

This problem is solved in three versions. 1) In the present form, when the length of any loop better than the present upper bound for empty/loaded vehicle travel lies in the above range. It is the range between the length of the loop corresponding to the best upper bound before the last trial, and the length of the loop in the last trial to find a better upper bound.

2) Without constraint (b) to find the shortest loop of a specific layout.

3) Where constraint (b) is in equality form, and the objective function is to either minimize or maximize the sum of a set of binary variables. That is to check whether the specific set of variables could all be equal to 0 or 1. The objective function \( Z_L \) of the Loop problem, whether in the first or second form, and its basic solution are added to the constraints of *Flow* problem. The new problem is called *Flow|Loop*. The objective of this problem is to find the MiniMax flow passed through a specific loop. The *Flow|Loop* problem is solved in a much shorter CPU time than the *Flow* problem which is relaxed in the loop. However, because of the symmetries and parallel lines in the block layout, usually the *Loop* problem has multiple optimal solutions. Therefore, not a specific loop, but a loop with some possible variations is transferred into the *Flow|Loop* problem. This is done through a set of constraints restricting the values of \( Y_{mn} \) variables\(^2\). For example, given all loops with equal length in a block layout, some edges are common in all of them, some are not on any, some pairs of edges could not be both on the loop, and so on.

\(^2\)The direction of the loop in the problem *Flow|Loop* is not known in advance
Problems Range of Covering Arcs \([RCA]\) is implemented to tighten the LP-relaxation of the \textit{Flow} problem to the Multiple loops of equal length. Problem \textit{Loop} in its third form is implemented as the sub-problem of these problems.

The RCA problem finds the minimum and maximum possible number of edges on a loop with a specific length.

\textbf{Problem RCA}

\[
\text{Minimize}(\text{Maximize}) Z_{RCA} = \sum_{mn \in A} Y_{mn}
\]

(a) Loop Configuration

\[
\sum_{mn \in A} Y_{mn} l_{mn} = L
\]

The following constraints are added to the problem \textit{Flow} \| \textit{Loop}.

\[
\sum_{mn} Y_{mn} \geq \text{Min } Z'_{RCA}
\]

\[
\sum_{mn} X_{mn} \leq \text{Max } Z'_{RCA}
\]

Now suppose that \(Y_B\) is the set of basic variables, and \(Y_{NB}\) is the set of nonbasic variables in the optimal solution of the problem \textit{Loop}\(^3\). It is useful to identify those sub-sets of basic variables which could not all become nonbasic, or zero basic in any of the corresponding loops. These are \(Y'_B \subseteq Y_B\) where their values in any feasible solution to the third form of the \textit{Loop} problem is greater than zero. Therefore,

\[
\sum_{Y_{mn} \in Y'_B} Y_{mn} \geq 1
\]

cuts a portion of the feasible region of the LP-relaxation not containing any loop with the length \(Z'_L\).

Similarly, the problem Covering Nonbasic Variables (CNV) identifies those sub-sets of nonbasic or zero basic variables which could not all become non-zero basic in any multiple optimal solution. These are \(Y'_{NB} \subseteq Y_{NB}\), where

\[
\sum_{Y_{mn} \in Y'_{NB}} Y_{mn} \leq |Y'_{NB}| - 1
\]

\(^3\)Note that the problem is extremely degenerate.
is also a constraint in any loop with the length \( Z^*_L \).

**Problem Flow|Loop**

\[
\text{Minimize} \\
Z_{F|L} = \text{Max} \{ \text{Flow passed through arcs of a loop with specific length} \}
\]

\[
s.t. \\
(a) \quad \text{Loop configuration} \\
(b) \quad \text{Length of the Loop} = Z^*_L \\
(c) \quad \text{Constraints generated by RCA} \\
(d) \quad \text{Material flow}
\]

**5.4.2 The Brute Force Search**

Step 1: The problem Loop in its first form is solved to find \( L_1 = Z^*_L \), then the problem \( Flow|Loop \) is solved to find \( F_1 = Z_{F|L_1}^* \). Similarly, the problems Flow and Loop|Flow are solved to find \( F_2 = Z^*_F \) and \( L_2 = Z_{L|F_2}^* \) respectively. The upper bound for the empty/loaded vehicle travel is the Min \( \{ (L_1 \times F_1), (L_2 \times F_2) \} \). These terms for the prototype example are \((100 \times 80)\) and \((118 \times 65)\) respectively. The upper bound here is 7670. If there exists a loop with smaller objective function, its length and flow lies between the above bounds.

Step 2: The Loop problem is solved to find a loop with the length \( L \) in the range \( L_1 + \epsilon \leq L \leq L_2 - \epsilon \). This is done either using the loop configuration portion of the model developed in this chapter, or preferably using the shortest loop model, Asef-Vaziri, et al. [1996]. This sub-problem does not search for multiple optimal solutions. If there does not exist any loop in the stated range, then the current upper bound is the optimal solution. Otherwise, the problems RCA is solved to generate additional constraints for the problem \( Flow|Loop \).

Solving the Loop problem for the prototype example shows that under the constraints of \( 101 \leq L \leq 117 \), there exist two loops of length 114 which are shown in
Figure 5.4: Loops with $100 < \text{length} < 118$

Figure 5.4. Note that in any loop of length 114, the lower edge of cell H must be equal to 1 the upper edge of cell G has to be 0, sum of the edges of cell M is $\geq 1$. Adding this constraints will lift the LP-relaxation of the Flow|Loop problem.

Step 3: The problem Flow|Loop is solved for the current loop. The maximum flow passed through a loop having a specific length of $Z_L^*$ is identified as $Z_{FIL}^*$. The problem Flow takes 2 hours CPU time for this prototype example. This time reduces to 10 minutes for the Flow|Loop problem. Furthermore, by fixing the direction of the loop, the problem is solved in 20 seconds. Therefore, it requires much less CPU time to solve the problem twice, each time for one direction, rather than solve it once for the undirected loop. The Minimax flow passing through an edge of a loop with length 114 is 80. The solution is shown in Figure 5.5. The total loaded/empty vehicle travel is 9120.

Step 4: Set $L_1 = \text{the length of the loop corresponding to the upper bound, and } L_2 = Z_L^*$. If $Z_L^* \cdot Z_{FIL}^*$ is less than the current upper bound, replace the upper bound with the new bound. Go to step 2. Since there is no other loop with its length greater than 100 and less than 114. The solution shown in Figure 5.2 is optimal.
### 5.5 Scope for future research

One possibility for future research would be to examine basis of the shortest loop model for two other configurations: 1) To find a Minimum Spanning Tree, covering at least one edge of each cell. 2) To find a Minimum Spanning Path with at least one edge in common with each cell. However, it is not known in advance which edges or nodes are covered by these specific type of spanning tree and path. Their difference with MST and MSP is the same as the difference between the shortest loop problem and TSP. Except that TSP is NP-Hard while MST is linear.

Another possibility for future research is to examine whether it is mathematically possible to show that the tendency towards forming sub-tours in this problem is not strong. If this could be possible, then instead of the enumeration process for generating potential sub-tours, a set of heuristics may be developed to identify a major set of primitive sub-tours. Furthermore, additional sets of intelligent Branch and Bound procedures may be developed to reduce the solution times for larger
problems.

A third possible research direction would be to customize the TSP heuristics for this problem, and to examine their efficiencies.

Regarding integration of flow loop and station locations, locating the stations on edges versus intersections may be examined.

Another direction is to develop a procedure to generate tandem configuration out of the single loop, or adding short cuts to the loop.

The followings are still some other directions for further research.

To apply heuristics like Simulated Annealing, Tabu Search, and Generic Algorithm to find a good solution for empty and loaded vehicle travel distance.

To develop a simulation model to compare four analytical solutions of loaded travel and stations on intersections, loaded travel and stations on edges, empty and loaded travel and stations on intersections, empty and loaded travel and stations on edges. The simulation model could also be utilized to design a short cut on a unidirectional loop.
Bibliography


Appendix A

The Potential Sub-tour Generation Process

For efficient generation of sub-tours, a generator matrix of size $|C| - 1$ is defined. Its first row is simply the row of cell $c$ in the adjacency matrix. Given row $i$ of the generator matrix, $i = 1, 2, \ldots, |C| - c$, whenever the process backtracks to row $i$, the flag of this row is moved to the next column $k$ for which $\text{Generator}(i, k) \neq 0$. Then row $i + 1$ of the generator matrix, for $j = \text{flag}(i - 1)$ to $C$, is derived as follows.

$$\text{Generator}(i + 1, j) = \max\{\text{Generator}(i, j), \max_{k \in C}\{\text{Adjacency}(k, j), \text{Adjacency}(j, k)\}\}$$

$$\text{Exclude}(j) = \max\{\min_{k \in C}\{\text{Adjacency}(k, j), \text{Adjacency}(j, k)\}\}$$

When generating the sub-tours initiated by cell $c$, the first row of the generator matrix identifies all potential 2-cell sub-tours $ck$, where $k > c$. When generating a specific 2-cell sub-tour $ck$, then the second row of the matrix identifies all 3-cell sub-tours initiated from $ck$. Row 3 generates all 4-cell sub-tours initiated from specific 3-cell sub-tour $ckl$, and so on. Note that for each child $cklm \ldots$, where all $k, l, m, \ldots > c$, there is no specific relationship among themselves. The algorithm shown in Figure A.1 finds all possible sub-tours.
Figure A.1: Potential Sub-Tour Identification Algorithm 1

Potential Sub-Tour Identification Algorithm 1

Begin
{
Set{TotalNumberOfSubTours, SubToursOfSize[j]=0}
For{all cell ∈ C)
  Set{Flag[j]=cell, CurRow[], NextRow[], Generator[]} = 0
Copy{ Adiacency Matrix AM[cell] into CurRow[]]
Set{check, chain=0, M=|C|-cell}
while{chain ≤ M}
  Set{STst[] = 0}
  If{check = 0
    Set{Generator[chain][] = 0}
    For{i > cell}
      Set{Generator[chain][i] = CurRow[i]}
  } else{
    Set{check, CurRow[] = 0}
    For{i ≥ flag[chain]}
      CurRow[i] = Generator[chain][i]
  }
  For{j ≥ flag[chain]}
    If{CurRow[j] ≠ 0}
      Set{flag[chain] = j, STst[0] = cell}
      For{i ≤ chain}
        Set{STst[i] = flag[i-1], NCells = chain + 2}
        Print{STst[]}
  }
Add{1 to SubToursOfSize[NCells], TotalNumberOfSubTours}
If{chain ≠ M}
  For{cell < i < to j}
    Copy{merged adjacency row and column of cell into}
    the NextRow[]
  For{i > cell}
    Merge{CurRow[i] into NextRow[i]}
  For{i ≤ chain}
    For{cell < ii ≤ flag[i]}
      If{Generator[i][ii] ≠ 0}
        Set{NextRow[ii] = 0}

Figure A.2: Potential Sub-Tour Identification Algorithm 2

**Potential Sub-Tour Identification Algorithm 2**

```
For{i <= cell
    CurRow[i]=0}
For{i > cell
    CurRow[i]=NextRow[i]}
    Break
}

If{j=|C|-1}
    Set{check=1, flag[chain]=cell, chain-1}
    If{chain<cell
        Break}
else{
    If{chain=M
        Set{check=1, chain-1}
    }
    Set{chain+1}
}
return
```
Appendix B

End Points of the Sub-tour Generation Process

The following algorithm identifies the termination point of each branch of each generation, and backtracks the enumeration process to the parent node.

Impossible Sub-Tour and Feasible Loop Identification 1

Begin

{  Read( all c ∈ s)
   For{ all c ∈ C₀
      Set{ flag(c) = |C₀|}
   }
   For{ all c ∈ in current sub-tour s
      For{ all k where ck ∈ A
         Set flag(k)=-1
      }
   }
   For{ all c ∈ s
      Set flag(c)=0}
   For{ all c ∈ s
      For{ all k where ck ∈ A
         flag(c) = flag(c) + flag(k)
   }
If(flag(c)==0
    back track (s is an impossible sub-tour)
}

For all $c \in C_o$
    flag(1) = flag(1) + flag(c)
If(flag(1) < 0
    back track (s is a feasible loop)
}
Appendix C

Checking Primitivity and Expanding Procedure

The pseudo-codes of the algorithm of checking for primitivity is shown below.

**Primitive Sub-Tour Identification 1**

1) Given a sub-tour $s$
   1) Find $z$ in $G_x$.
   2) Find $\exists = C\backslash (s + z)$.
   3) If $|\exists| < |s|$, Then $s$ is not primitive. Stop.
   4) Find $r$ as the set of nodes in the smallest component of $G[\exists]$.
   5) If $|s| > |r|$, The potential sub-tour is not primitive. Stop.
   6) If $|s| = |r|$, find $c$ and $k$ as the smallest identification number among cells in $s$ and $r$ respectively. If $k > c$, the proposed sub-tour is not primitive. Stop.
   7) Go to expanding procedure.

Details of checking for primitivity and expanding the elimination constraints are shown in the following algorithm.

**Primitive Sub-tour Identification and Expanding 1**

*FindPrimitiveSubTours*

*Begin*

*Given { sub-tour s}*

159
Copy\{G and G_\times \text{ into } G^1 and G^1_\times \}
Find\{s as the set of cells adjacent to s in G^1_\times\}
Set\{s = C \backslash (s + s)\}

If\{|s| < |s|
\quad s \text{ is not primitive. Stop}\}.

Set\{MinC_1 = \infty\}
Find\{}MinS1= \text{The smallest identification number among } c \in s\Find\text{Induce}\{G^1[s]\}
\text{While}\{
\quad \text{If}\{\text{there is a remaining cell } c_1 \text{ in } s\}
\quad \quad \text{Find}\{\text{the component } C_1 \text{ containing } c_1 \text{ in the induced graph } G^1[s] \}\}
\quad \text{else}\{\text{Break}\}
\quad \text{If}\{MinC_1 > |C_1|\}
\quad \quad \text{Set}\{MinC_1 = |C_1|\}
\quad \text{If}\{MinC_1 < |s|
\quad \quad s \text{ is not primitive. Stop}\}
\quad \text{If}\{|C_1| = |s|
\quad \quad \text{Find}\{\text{MinS2= The smallest identification number among } c \in C_1\}
\quad \quad \text{If}\{MinS1 > MinS2\}
\quad \quad \quad s \text{ is not primitive. Stop.}\}
\quad \text{Copy}\{G \text{ and } G_\times \text{ into } G^2 \text{ and } G^2_\times\}
\text{Find}\{s' \text{ as the cells adjacent to } s \text{ in } G^2_\times\}
\text{Find}\{\text{LXS and SXS as the largest and smallest components of } G^2[C \backslash (s + s')]\}
\quad \text{If}\{|LXS| > |C_1| \text{ or } |SXS| < |C_1|
\quad \quad s \text{ is not primitive. Stop.}\}\}
\quad \text{Set}\{s = s \setminus C_1\}\}
\text{If}\{MinC_1 > |s|
Copy\{G and G_x into G^1 and G_x^1 \}
Set\{\bar{s} = C\backslash(s + \bar{s})\}
Find\{s' as the cells adjacent to \bar{s} in G_x^1\}
Induce\{ G^1[C\backslash(\bar{s} + s')]\}
Find\{XS as the size of the largest component of G^1[C\backslash(\bar{s} + s')] containing s\}
If\{XS < MinC_1
        MinC_1 = XS\}
Add\{1 to the number of primitive sub-tours of size MinC_1\}