TECHNICAL RESEARCH REPORT

Minimization of Acquisition and Operational Costs in Horizontal Material Handling System Design

by J.W. Herrmann, G. Ioannou, I. Minis, J.M. Proth

T.R. 95-12

INSTITUTE FOR SYSTEMS RESEARCH

*Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry*

Minimization of acquisition and operational costs in horizontal material handling system design

J.W. Herrmann* G. Ioannou† I. Minis† J.M. Proth‡

Abstract

This paper considers the problem of minimizing the fixed cost of acquiring material handling transporters and the operational cost of material transfer in a manufacturing system. This decision problem arises during manufacturing facility design, and is modeled using an integer programming formulation. Two efficient heuristics are developed to solve it. Computational complexity, worst-case performance analysis, and extensive computational tests are provided for both heuristics. The results indicate that the proposed methods are well suited for large-scale manufacturing applications.

Keywords: Material Handling Systems, Manufacturing Systems,Vehicle Routing

*Institute for Systems Research, University of Maryland, College Park, MD 20742

†Dept. of Mechanical Eng. and Inst. for Systems Research, Univ. of Maryland, College Park, MD 20742

‡INRIA-Lorraine, 4 Rue Marconi, Metz 2000, 57070 Metz, France

1 Introduction

Material handling contributes significantly to overall manufacturing costs. Fixed costs are associated with the investment in material handling equipment during system construction, while variable costs arise from the material transfer between the resources of the manufacturing facility during system operation. These costs are conflicting in nature, since acquiring more equipment may reduce the material handling effort at the expense of increased investment. The goal of this study is to support the material handling system design process by determining the minimum number of transporters required to transfer parts/batches in the manufacturing facility with minimal material handling effort

Most of the related research discussed below has addressed Automated Guided Vehicle (AGV) systems. However, it is equally applicable to all types of horizontal, carrier-based transportation systems such as rail carts, industrial trucks, forklifts etc. [10].

Egbelu [6] developed a set of simple formulas to calculate the minimum number of carriers in a manufacturing system, based on the loaded traveling times as well as on some empirical estimates of the unloaded traveling times. This work can be employed in an initial economic justification of AGV systems. To address the same problem, Tanchoco *et al.* [18] employed a queueing theory-based computer model (CAN-Q), and Wysk *et al.* [19] used spread-sheet analysis. Their results compared favorably to a simulation-based method (AGVSim). All these approaches provide initial estimates for the number of carriers, which may be further refined by simulation. As such, they do not consider detailed aspects of the problem, that may be important during system operation, such as the distribution of moves (loaded and unloaded) among the vehicles. For the problem of determining the AGV fleet size, Sinriech and Tanchoco [16] developed a multi-criteria optimization model that considers the trade-off between investment costs and system throughput. To support the design process they proposed the use of decision tables relating the investment cost, the number of AGVs and their utilization, as well as the trade-off ratio between the corresponding conflicting costs. In this approach, optimality is not guaranteed and the deviation from the optimal of a solution chosen by the designer cannot be evaluated.

The design of efficient horizontal unit-load material handling systems was studied by Maxwell and Muckstadt [13]. They considered the case in which the production rate of each manufacturing resource is constant and knowne. To determine the minimum number of AGVs, they solved a transportation problem that distributes the unloaded vehicle moves among pairs of resources, in a way that minimizes the unloaded vehicle traveling time. Subsequently, they determined sequences of moves that originate and terminate at the same resource (*routes*) and assigned them to AGVs. Their routing algorithm created a delivery schedule with a near-constant inter-arrival time of material at each resource. The authors, however, do not provide analytical tools to determine vehicle routes, which are critical in assigning moves to vehicles. In large applications, evaluation of these routes may be a complex, and often intractable, problem. Furthermore, the authors do not consider the periodicity of vehicle operations that may lead to additional unloaded moves in order to drive the system to its initial state.

In a related research area, the merit of topologically simple flow path designs for better AGV control has been examined by several authors (e.g. Bartholdi and Platzmann [1], Bozer and Srinivasan [2] and Sinriech and Tanchoco [17]). Bozer and Srinivasan [3] presented a partitioning algorithm for the design of single vehicle loops, in an effort to distribute the workload evenly among the AGVs in the material handling system. Although these designs offer simplicity and allow analytical performance evaluation, no rigorous arguments have been presented to support their advantages against conventional networks.

Assuming a fixed shop layout with predetermined material flow paths, the problem of minimizing fixed acquisition and variable transportation costs addresses the following control-related issues: i) Assignment of available transportation equipment to service requests by jobs waiting in the queues of output stations, and ii) transporter optimal routing from a resource output station to the destination input station. However, since this problem is relevant to the design stage of a manufacturing facility, during which no real-time information is available, we propose a static integer programming formulation closely related to vehicle routing [7]. The resulting optimization problem is *NP*-hard, and two efficient heuristics are developed to solve it. The first is a greedy algorithm similar to the nearest neighbor approach for the traveling salesman problem. The second is a composite algorithm that solves the assignment relaxation of the integer program, determines closed sequences of resources with a common origin (routes) and assigns these routes to transporters by solving a two-stage bin-packing problem [11, 12]. Both heuristics run in polynomial time and their worst case performance is bounded by ratios of problem parameters. Extensive computational tests against lower bounds also show that they provide satisfactory solutions for applications of practical size.

The remainder of this paper is organized as follows. Section 2 introduces our assumptions as well as relevant definitions and notation. Section 3 presents the integer programming formulation of the problem and the assignment model used to compute lower bounds. Section 4 describes the two heuristic solution approaches, and Section 5 presents results on the computational complexity and the worst case performance of the heuristics. Section 6 includes the numerical experiments, and Section 7 summarizes the conclusions of this work.

2 Assumptions, definitions and notation

The development of the mathematical model is based on the following assumptions: 1) The placement of the manufacturing resources on the shop floor is given, together with the location of the resource pick-up (output) and drop-off (input) stations. (For a review of effective shop layout techniques see [8, 14].) 2) The material flow paths between resources are fixed. (For existing methods in flow path design see [9].) 3) The inter-resource material flow rates (in terms of loads per unit time) are constant from time period to time period and known. They are calculated from the production routings (sequences of operations) of the products to be manufactured and their demand over the design horizon. 4) Whenever a transporter visits a resource output station, there always exists material to be transferred to subsequent resources. This assumption is necessary since no real-time information is available at the system design stage. 5) Horizontal material handling transporters are considered (e.g. AGVs, manual or automated rail carts, industrial trucks, and forklifts) with unit load capacity; no sharing of moves between different material flow types (i.e., different batches) is allowed.

There exist three types of transporter operations between a pair of manufacturing resources: i) A *loaded move* i is the transporter operation from the output station of the manufacturing resource $o(i)$ to the input station of the destination resource, $d(i)$. The set of loaded moves is denoted by L, and its cardinality ($|L|$) will be referred to as n throughout the text. ii) An *unloaded move* is the transporter operation from the input station of a manufacturing resource to the output station of another resource, during which no load is carried. The set of all possible unloaded moves is denoted by U. iii) A *complete move* is the concatenation of a loaded move and a subsequent unloaded move. The set of complete moves is denoted by C.

Each element of L is associated with a unit entry of the flow matrix. Assuming that there exists a path between each input-output pair, it is easy to see that after the completion of a loaded move, a transporter can perform an unloaded move to the origin of any other loaded move in L. As a result, either three or only two resources may be included in a complete move. In the latter case,

$d(i) = o(j)$, where i and j are consecutive moves.

A cost c_{ij} is associated with each complete move $(i,j) \in C$. Assuming that the costs τ_i, τ_i^p, τ_i^d, and τ_i^j reflect the time needed to perform loaded move i, to pick-up the load from $o(i)$, to deliver the load to $d(i)$, and to travel from $d(i)$ to $o(j)$, respectively, then c_{ij} is defined as:

$$c_{ij} = \begin{cases} \tau_i + \tau_i^p + \tau_i^d + \tau_i^j & \text{if } i \neq j \\ \infty & \text{if } i = j \end{cases} \tag{1}$$

The set of material handling transporters available for transfer of parts is denoted by V. For each transporter $k \in V$, the scaled capital investment is denoted by w_k; this cost is appropriately scaled to reflect the relative weights of the variable and fixed components of the objective function.

A *route* u is a sequence of moves performed by a transporter that originates and terminates at the same resource output station. This definition is adopted from Maxwell and Muckstadt [13], and will be employed by the second heuristic presented in Section 4. A set of routes that have the same origin is a *route set*, denoted by Γ. The time needed to perform all the complete moves of a route is denoted by $c_u = \sum_{(i,j)\in\gamma_u} c_{ij}$, where γ_u is the set of complete moves of route u. In the remainder of the paper we will refer to a route either by its index u or by the corresponding set γ_u.

Finally, T is the period within which all loaded moves must be performed. Note that T is scaled appropriately to reflect the time costs c_{ij} in (1).

3 Mathematical model

To formulate the problem of minimizing the fixed acquisition and the variable operational costs of the material handling system, we use the following additional notation: For all elements of the transporter set V we define a binary variable that indicates which transporters perform at least one loaded move in L, and thus should be acquired; i.e.,

$$y_k = \begin{cases} 1 & \text{if transporter } k \in V \text{ is employed for some move in } L \\ 0 & \text{otherwise} \end{cases}$$

Let x_{ij}^k be a binary variable associated with each complete move:

$$x_{ij}^k = \begin{cases} 1 & \text{if move } j \in L \text{ is performed following move } i \in L \text{ by transporter } k \in V \\ 0 & \text{otherwise} \end{cases}$$

The design problem can now be expressed as follows:

Problem $\mathcal{P}$

$$\text{minimize} \qquad Z = \sum_{k \in V} w_k y_k + \sum_{(i,j) \in C} \sum_{k \in V} c_{ij} x_{ij}^k \tag{2}$$

subject to :

$$\sum_{(i,j) \in C} \sum_{k \in V} x_{ij}^k = 1 \qquad \forall i \in L \tag{3}$$

$$\sum_{(i,j) \in C} \sum_{k \in V} x_{ij}^k = 1 \qquad \forall j \in L \tag{4}$$

$$\sum_{(i,j) \in C} x_{ij}^k - \sum_{(j,i) \in C} x_{ji}^k = 0 \qquad \forall i \in L, \quad k \in V \tag{5}$$

$$\sum_{i \in F} \sum_{j \in F} x_{ij}^k \leq \sum_{i \in F} \sum_{j \in L} x_{ij}^k - 1 \qquad \forall F \subset L : 2 \leq |F| < \sum_{(i,j) \in C} x_{ij}^k, \quad \forall k \in V \tag{6}$$

$$\sum_{(i,j) \in C} c_{ij} x_{ij}^k \leq T \qquad \forall k \in V \tag{7}$$

$$x_{ij}^k \leq y_k \qquad \forall (i,j) \in C, \quad k \in V \tag{8}$$

$$x_{ij}^k, y_k \in \{0,1\} \qquad \forall (i,j) \in C, \quad k \in V \tag{9}$$

The objective function Z in (2) accounts for the capital investment to acquire transportation equipment and for the operational cost of material handling. Constraints (3)-(4) ensure that each loaded move in L is performed by exactly one transporter $k \in V$. Constraint (5) imposes continuity on the path consisting of loaded moves performed by each transporter, by ensuring material flow conservation. The exponential set of constraints (6) enforces subtour elimination, guaranteeing the existence of a single tour for each transporter $k \in V$. Note that a subtour is a sequence of moves of the form $(i_1, i_2, ..., i_1)$. Constraint (7) limits the time that each transporter operates to the design horizon T. Constraint set (8) prohibits material movement by non-activated vehicles ($y_k = 0$). Finally, constraint (9) forces the variables x_{ij}^k and y_k to assume binary values.

In order to guarantee a feasible solution to problem $\mathcal{P}$, the following property should be satisfied by each complete move: $c_{ij} < T/2, \forall (i,j) \in C$. The case in which there exist some complete moves with $c_{ij} > T/2$ may lead to an empty feasible solution space, if these moves have to be performed by the same transporter. Based on this property, and the fact that at least two loaded moves are required to complete a tour, it is easily seen that

$$\sum_{k \in V} y_k \leq \lceil \frac{n}{2} \rceil \tag{10}$$

Assignment lower bounds

The problem that results by considering only the operational cost in the objective function of $\mathcal{P}$, by removing the capacity constraints (7), and by disregarding the transporter indices, is the well known assignment problem (or minimum weight matching problem [15]) which is presented below.

Problem $\mathcal{A}$

$$\text{minimize} \qquad Z_a = \sum_{(i,j)\in C} c_{ij}x_{ij} \tag{11}$$

subject to :

$$\sum_{(i,j)\in C} x_{ij} = 1 \qquad \forall i \in L \tag{12}$$

$$\sum_{(i,j)\in L} x_{ij} = 1 \qquad \forall j \in L \tag{13}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in C \tag{14}$$

The optimal solution Z_a^* of $\mathcal{A}$ provides for every loaded move $i \in L$ the loaded move $\phi(i)$ that should follow i, such that the total variable cost is minimized. Z_a^* and the associated $\phi(i)$'s are derived in polynomial time by the Hungarian algorithm, an application of the primal-dual method [15]. It is clear that Z_a^* bounds from below the variable component Z_{opt}^v of the optimal solution of $\mathcal{P}$, i.e. $Z_a^* \leq Z_{opt}^v = \sum_{(i,j)\in C}\sum_{k\in V} c_{ij}x_{ij}^{k^*}$, where $(x_{ij}^{k^*}, y_k^*)$ is the optimal variable vector of $\mathcal{P}$.

The above lower bound on the variable cost can be used to obtain a lower bound on the number of transporters, which is given by:

$$R_a^* = \lceil \frac{Z_a^*}{T} \rceil \tag{15}$$

Note that if we assume identical transporters in terms of acquisition costs, i.e. $w_k = w$, $\forall k \in V$, then we have a lower bound on the optimal fixed cost in (2): $R_a^* \cdot w \leq Z_{opt}^f = \sum_{k\in V} w_k y_k^* = w\sum_{k\in V} y_k^*$. The lower bounds derived above will be employed for the evaluation of the heuristics during numerical experiments.

4 Solution algorithms

Since problem $\mathcal{P}$ is closely related to the well known *NP*-complete vehicle routing problem, optimal solutions cannot be computed for medium to large-sized problems. Thus, we develop heuristic approaches that provide near-optimal solutions.

4.1 A greedy heuristic

Let us consider the set of loaded moves. An intuitive approach for minimizing the cost Z is to start matching moves that have the minimum cost coefficients c_{ij}, and attempt to allot as many complete moves to each available transporter as possible. Note that by matching two loaded moves $i, j \in L$, an unloaded move from $d(i)$ to $o(j)$ is fixed. Thus, we can proceed by allotting to each transporter complete moves with minimal cost in a greedy fashion, until capacity constraints are violated. This results in selecting minimum cost moves at the beginning of the procedure. However, as the algorithm proceeds, non-favorable selections may be made, as is typical with nearest-neighbor type approaches. The minimization of the number of transporters is implicitly introduced by forcing each transporter to be loaded to near-capacity before another transporter is activated. This greedy algorithm is presented below. Note that L_g is an auxiliary set used in the presentation of the algorithm, k is the transporter index, and D_k the associated variable cost.

Algorithm *GREEDY*

1. Set $k = 1$, $L_g = L$
2. If $L_g = \emptyset$, go to 8
3. Choose a loaded move $p \in L_g$ at random
4. Set $q = p$, $D_k = 0$
   $L_g = L_g \setminus \{p\}$
5. Select $j \in L_g : c_{pj} = min_{j \in L_g}\{c_{pj}\}$
6. If $D_k + c_{pj} + c_{jq} \leq T$, then:
   $D_k = D_k + c_{pj}$
   $L_g = L_g \setminus \{j\}$
   If $L_g = \emptyset$ then
   $D_k = D_k + c_{jq}$
   Go to 8
   $p = j$
   Go to 5
7. If $D_k + c_{pj} + c_{jq} > T$, then
   $D_k = D_k + c_{pq}$
   $k = k + 1$
   Go to 2
8. Output number of transporters, assignment of moves, and variable cost

The above algorithm is straightforward. Capacity constraints play an important role in its progress, since they impose the threshold for transporter activation and, thus, for fixed cost allocation. *GREEDY* attempts to assign complete moves of minimum cost to transporters; at the same time it forces each vehicle to perform closed continuous loops. Note that regardless of the value of the fixed costs, the above heuristic will provide the same solution for a given set of variable costs.

4.2 An assignment/bin-packing (ABP) composite heuristic

An intuitive first step towards minimization of the variable portion of Z in (2) is to solve the assignment problem $\mathcal{A}$. Starting from any loaded move $i \in L$ and following the sequence $\phi(i), \phi(\phi(i))$, ..., we will return to i, since $\exists j_1 \in L : \phi(i) = j_1$ and $\exists j_2 \in L : i = \phi(j_2)$. The resulting closed sequences of loaded moves form subtours. If one transporter is allotted to each subtour, then the variable cost in (2) is minimal. However, this *ad-hoc* allotment of moves to transporters may not be feasible -since capacities may be violated- and is, in general, not economical -since an unnecessary large number of transporters may be activated. Consequently, an algorithmic approach driven by the preservation of the minimal variable cost is required to translate the solution of $\mathcal{A}$ into a near-optimal solution of $\mathcal{P}$. This is the basis of the heuristic ABP presented below.

ABP starts with the subtours of the optimal solution of $\mathcal{A}$ and allots moves to transporters in order to minimize the objective of $\mathcal{P}$. This is accomplished by identifying routes and routes sets and performing a two-stage bin-packing [11, 12]. Figure 1 illustrates the definition of subtours, routes, and route sets. A subtour derived from the solution of the assignment problem is shown in

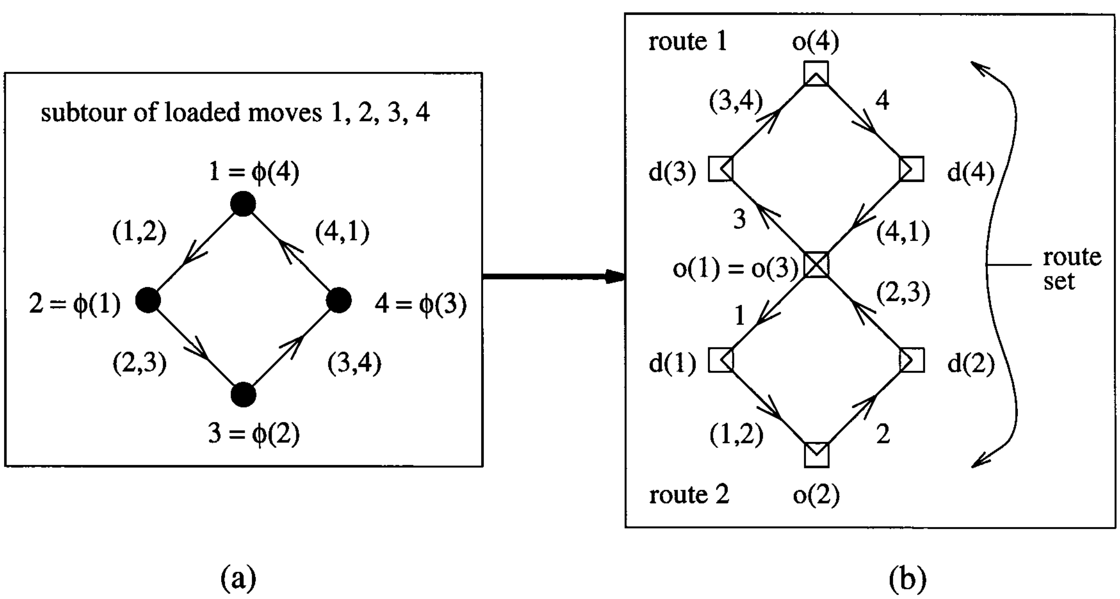

Figure 1: Illustration of terminology: a) moves and subtours; b) routes and route sets generated by the subtour of (a)

Figure 1a. The nodes $i = 1, ..., 4$ represent loaded moves, while the arcs $(1,2), ..., (4,1)$ represent the unloaded moves connecting the loaded ones. If some moves of the subtour have common origin, then the subtour is decomposed into routes which intersect at this origin. This transformation of moves to routes is unique, since it is based on the assignment matchings. In Figure 1b, two routes are shown, connected at $o(1) = o(3)$. These routes form one route set corresponding to station $o(3) = o(1)$. The nodes of a route set represent resource input/output stations, while the arcs represent loaded/unloaded moves.

In the algorithm given below, L_b is an auxiliary set initially equal to the set of loaded moves; $pack(i)$, $\forall i \in L$ is the transporter that move i is assigned to; K is the total number of transporters; D_k is the total scaled time it would take transporter k to perform the moves allotted to it; t_u is a variable that corresponds to each activated transporter; M is the set of resources and f_{rs} the material flow between resources $r, s \in M$. Furthermore, FFD refers to the first-fit-decreasing algorithm for solving bin-packing problems; the FFD heuristic was employed due to its tight worst-case bounds (11/9 times the optimal [11]). Finally, the load of a transporter is the sum of the cost coefficients associated with the complete moves performed by this transporter.

Algorithm ABP

1. Solve $\mathcal{A}$ to obtain $\phi(i)$, $\forall i \in L$
2. Set $L_b = L$
3. Set $u = 1$, $v = 1$, $\gamma_u = \emptyset$, $c_u = 0$, $\Gamma_v = \emptyset$
4. Identify $r \in M : \sum_{s \in M} f_{rs} = max_{m \in M}\{\sum_{s \in M} f_{ms}\}$
5. Select move $i \in L_b : o(i) = r$
6. do
    $L_b = L_b \setminus \{i\}$
    $f_{o(i)d(i)} = f_{o(i)d(i)} - 1$
    $\gamma_u = \gamma_u \cup \{(i, \phi(i))\}$, $c_u = c_u + c_{i,\phi(i)}$
    $i = \phi(i)$
   until $o(i) = r$
7. $\Gamma_v = \Gamma_v \cup \{\gamma_u\}$
8. If $\sum_{s \in M} f_{rs} > 0$
    $u = u + 1$, $\gamma_u = \emptyset$, $c_u = 0$
    Go to step 5 (new route)
9. If $L_b \neq \emptyset$
    $v = v + 1$, $\Gamma_v = \emptyset$
    $u = u + 1$, $\gamma_u = \emptyset$, $c_u = 0$
    go to 4 (new route set)
10. For $\beta = 1, ..., v$, apply FFD to c_u's of $\gamma_u \in \Gamma_\beta$
    Get temporary transporters K, loads $D_1, ..., D_K$, and $pack(i), \forall i \in L$
11. Renumber transporters in decreasing order of D_k's, as $l = \{t_1, ...t_K\}$
12. Get first element of l
13. Set $p = q = 1$ and $l = l \setminus \{t_p\}$

14. do

Let t_q = next element of l, after current t_q

If $D_p + D_q \leq T$ then:

$\delta_{min} = min_{i \in t_p, j \in t_q}[c_{i,\phi(j)} + c_{j,\phi(i)} - c_{i,\phi(i)} - c_{j,\phi(j)}]$

If $D_p + D_q + \delta_{min} \leq T$ and $\delta_{min} \leq w_k$ then:

i) Set $l = l \setminus \{t_q\}$, $D_p = D_p + D_q + \delta_{min}$

ii) $\forall i \in t_q$, $pack(i) = t_p$

If t_q = last element of l:

Set t_p = first element of l

Set $l = l \setminus \{t_p\}$

Set t_q = first element of l

until $l = \emptyset$

15. Output transporter number and final assignment of moves to transporters

The solution of $\mathcal{A}$ generates the loaded move matchings $\phi(i)$ and the optimal subtours of loaded moves in Step 1 of algorithm ABP. After the initializations in Steps 2 and 3, routes and route sets are identified in Steps 4 through 9. Specifically, Step 4 determines the resource r with the maximum outgoing flow in the current flow matrix. Step 5 selects a loaded move i that originates at resource r. Step 6 follows the sequence of moves $i, \phi(i), \phi(\phi(i)), ...$, until a new move originating at r is encountered. This sequence forms route γ_u. Each time the loop of Step 6 is performed, the inter-resource flow matrix and the route cost c_u are appropriately updated.

Step 7 adds the route γ_u that has just been formed to the current route set Γ_v, which contains the routes that originate from the output station of resource r. If there remain moves that originate at resource r, Step 8 initiates a new route from this resource. Otherwise, Step 9 initializes a new route set, provided that there exist moves yet unassigned to routes ($L_b \neq \emptyset$). Steps 4 through 9 are repeated until all moves are assigned to routes and routes are grouped into route sets.

Routes are alloted to transporters in two stages. Step 10 solves a bin-packing problem for every route set Γ_v, in which the routes γ_u correspond to objects with weights equal to the route costs c_u and the bin capacity equals the time horizon T. The first-fit-decreasing (FFD) [11] algorithm is employed to determine the minimal number of bins (transporters) required to complete the moves in Γ_v. After this procedure is applied to all route sets Γ_v, a temporary assignment of routes (and moves) to K transporters is determined. Note that the optimality of the variable cost is maintained until the end of Step 10. Step 11 renumbers the temporary transporters in list l such that if $t_{k_1}, t_{k_2} \in l : k_1 < k_2$, then $D_{k_1} \geq D_{k_2}$.

The final stage of the algorithm addresses the fixed component of the objective function. Given the temporary assignment of routes to transporters, Steps 12 through 14 reduce the number of transporters by combining the routes assigned to more than one transporters. The goal is to reassign routes from underutilized transporters to more utilized ones, thus merging several disjoint routes not necessarily from the same route set. To achieve that, a procedure similar to the FFD heuristic is used. Step 13 removes the top element from the list l, i.e. the transporter with maximum load t_p, and Step 14 examines whether this transporter can be merged with any other one, without violating capacity constraints or increasing the variable cost more than the transporter fixed cost. The routes of the first subsequent (in l) transporter t_q that satisfies these conditions, are merged with those of t_p, to form a modified transporter t_p. In this case, the number of transporters is decreased by one, and t_q is removed from l. When all the remaining elements of the list are examined for possible merger with t_p, the algorithm returns to the top of l and examines possible transporter mergers with the next maximum load transporter. Step 14 is repeated until no further transporter mergers can be performed.

To calculate the minimum increase of the variable cost when attempting to merge the routes of two transporters, Step 14 calculates the cost augmentations for all possible connections δ_{ij} between moves i and j that are allotted to different transporters. The connection that results in the minimum cost increase is implemented. Figure 2 illustrates the process of cost augmentation for moves 2 and 3 that belong to routes $\{(1,2),(2,1)\}$ and $\{(3,4),(4,3)\}$. When attempting to

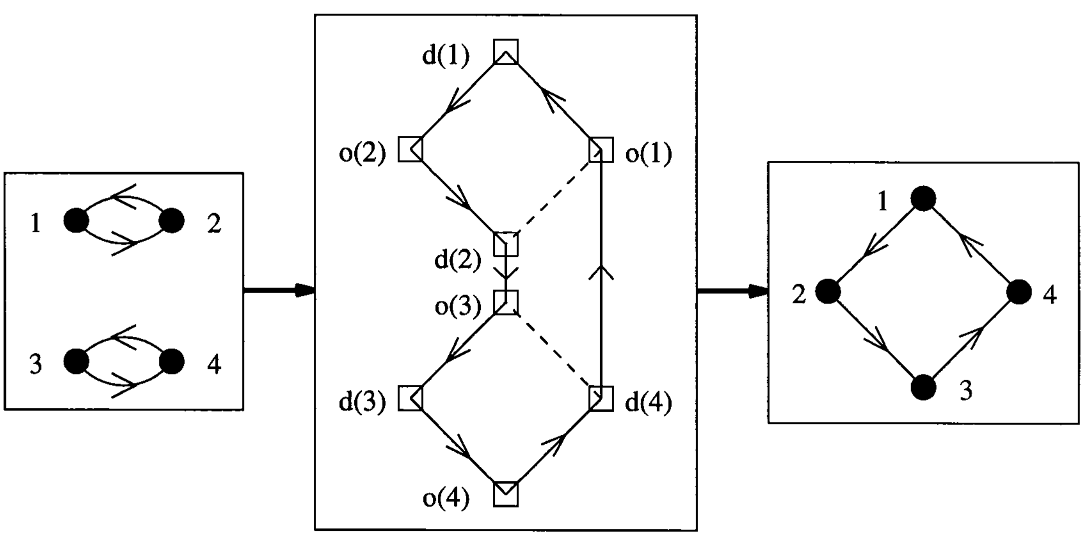

Figure 2: Cost augmentation during merging of transporter loads

merge the two routes, unloaded moves (2,1) and (4,3) are replaced by unloaded moves (2,3) and (4,1). This results in a variable cost increase of $\delta_{23} = c_{2,3} + c_{4,1} - c_{2,1} - c_{4,3}$. If this is the minimum cost augmentation (i.e., $\delta_{23} \leq \delta_{24}$, $\delta_{23} \leq \delta_{41}$, $\delta_{23} \leq \delta_{13}$) then the route connection is implemented between $d(2), o(3)$ and $d(4), o(1)$. After the route merger, one transporter will perform all moves 1, 2, 3, and 4. Note that in Step 14 optimality with respect to the variable cost may be sacrificed. However, by selecting the minimum augmentation cost at each iteration, the algorithm tries to keep the variable cost as close to the lower bound Z_a^* as possible.

It easy to see that *ABP* attempts to reduce the fixed cost in Steps 10 and 14, when routes from the same or different route sets are connected. However, it is clear that the main concern of algorithm *ABP* is the variable cost, the optimality of which is preserved until Step 14. Also, note that in order to apply the FFD in Step 10, we have assumed that $c_u \leq T$, $\forall u$. This assumption is well justified in manufacturing applications, since the moving and loading/unloading times are negligible compared to the design horizon T, resulting in small route costs.

5 Evaluation of heuristics

In this section some important properties of the two heuristics are established. The proofs of all theorems are included in Appendix A.

Computational complexity

The greedy heuristic is particularly fast; its computational complexity is bounded by a low order polynomial. The computational complexity of the composite assignment/bin-packing heuristic is dominated by the time to compute the solution to the assignment problem.

Theorem 1 *The computational complexity of GREEDY is $O(n^2)$.*

Theorem 2 *The computational complexity of ABP is $O(n^3)$.*

Worst case analysis

In this analysis we examine the variable and fixed costs separately. Let $c_{max} = \max_{(i,j) \in C} \{c_{ij}\}$, and $c_{min} = \min_{(i,j) \in C} \{c_{ij}\}$, i.e. the maximum and minimum elements of the cost matrix $[c_{ij}]$. Also, let Z_g^v and Z_b^v be the variable costs of the solutions derived by applying *GREEDY* and *ABP*, respectively, and R_g and R_b the corresponding numbers of the activated transporters.

The following two theorems establish the worst case performance of the heuristics, with respect to the variable cost.

Theorem 3 *For any instance of problem $\mathcal{P}$, algorithm GREEDY provides a solution that satisfies the property*

$$\frac{Z_g^v}{Z_{opt}^v} < \frac{c_{max}}{c_{min}}$$

Thus, if the ratio of the maximum to the minimum entry of matrix $[c_{ij}]$ is small, it is guaranteed that this heuristic performs well even in the worst case, with respect to the variable portion of the cost. In manufacturing systems, the transporter moves are bounded in the area of the shop floor. Thus, the move time costs are relatively close and the worst-case bound of Theorem 3 is anticipated to be particularly tight.

Theorem 4 *For any instance of problem $\mathcal{P}$, algorithm ABP provides a solution that satisfies the property*

$$\frac{Z_b^v}{Z_a^*} \leq \frac{c_{max}}{c_{min}}$$

Since $Z_{opt}^v \geq Z_a^*$, it follows directly from Theorem 4 that:

Corollary 1 *For any instance of problem $\mathcal{P}$, algorithm ABP provides a solution that satisfies the property*

$$\frac{Z_b^v}{Z_{opt}^v} \leq \frac{c_{max}}{c_{min}}$$

Let us now consider the fixed cost, which is proportional to the number of activated transporters if $w_k = w, \forall k \in V$.

Theorem 5 *For any instance of problem $\mathcal{P}$, algorithms GREEDY and ABP provide solutions that satisfy the property*

$$\frac{R_h}{R_{opt}} \leq \frac{1}{2} \cdot \frac{T}{c_{min}}$$

where R_h represents the number of transporters in the appropriate heuristic ($h = g$ for GREEDY, $h = b$ for ABP) and R_{opt} the number of transporters in the optimal solution.

Thus, although both heuristics are guaranteed to provide solutions for which the fixed part of the cost in (2) is bounded with respect to the optimum, they may perform arbitrarily bad, if the ratio of the time period T over the minimum entry of matrix $[c_{ij}]$ becomes large. However, the results of the next section indicate that on the average the numbers of transporters provided by the heuristics are very close to the optimal.

6 Numerical results

Algorithms *GREEDY* and *ABP* were implemented in the *C* programming language on a Sun Sparc workstation. In order to evaluate their effectiveness, a large number of sample problems of various sizes were solved. An algorithmic approach was employed to generate the random example sets, solve the resulting problems $\mathcal{P}$, and compare the solutions to the lower bound Z_a^*.

Each example set was generated by considering a square of side length b_i, with $1 \leq i \leq e$, $e \in \mathcal{N}^+$, as shown in Figure 3. The number of manufacturing resources $|M|$ was selected at

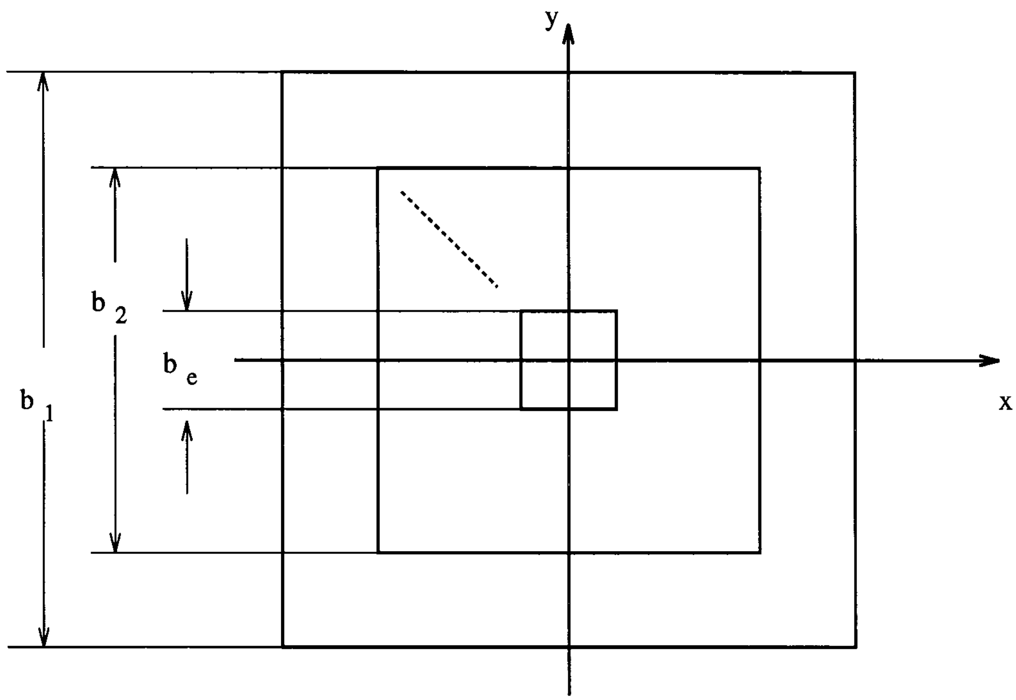

Figure 3: Example generation scheme

random, and the $x - y$ coordinates of the input and output stations of each resource were randomly generated within this square. Given these coordinates, the from-to distance matrix was evaluated. As a measure of the inter-resource distance we used the Manhattan distance, defined as: $\mu_{rs} = |x_r - x_s| + |y_r - y_s|, \forall r, s \in M$ The resulting distances were transformed to traveling times for a given transporter speed. The pick-up and drop-off times were assumed negligible. Based on the above assumptions, all the time costs for the formulation of problem $\mathcal{P}$ were computed.

For each pair of resources $r, s \in M$ the material flow intensity f_{rs} was randomly generated. However, the overall loaded traveling time, $\sum_{r \in M} \sum_{s \in M} f_{rs}$, was restricted between a lower and an upper bound, to generate examples with almost invariant loaded traveling time. Finally, a constant time horizon T, and constant fixed transporter costs were assumed.

For the square shop in Figure 3 with side b_1, a set of 100 different examples was generated. For each example, *GREEDY* was applied 20 times starting from different loaded moves and the best solution obtained was used in the comparisons. *ABP* is deterministic and, therefore, was applied to each example once. Subsequently, the dimension of the square shop was decreased by a factor σ (i.e., $b_i = \frac{b_{i-1}}{\sigma}$, $2 \leq i \leq e$), the average of the entries of the from-to flow matrix was increased by σ, and the example generation and solution procedure was repeated e times.

To gain some insight in the relative average performance of the two heuristics, the average results for $e = 10$, $T = 500$, and $\sigma = 1.5$ are shown in Table 1. The table includes the average

Table 1: Results of heuristics for 10 example sets
(average values over 100 examples per set)

		Variable cost		*Transporters*		*% Unutilized capacity*	
Example set	*No. of moves*	$\bar{Z}_g^v$	$\bar{Z}_b^v$	$\bar{R}_g$	$\bar{R}_b$	$\bar{lc}_g$	$\bar{lc}_b$
1	107.45	4035.38	3837.45	9.28	9.85	21.15	24.32
2	141.68	4012.27	3803.58	9.17	9.36	18.24	21.37
3	205.31	4001.55	3785.13	8.92	9.18	14.32	17.56
4	258.42	3967.48	3746.76	8.34	9.03	11.56	14.28
5	315.57	3943.94	3728.31	8.25	8.78	9.12	12.03
6	412.28	3917.32	3705.67	8.16	8.69	7.23	10.64
7	547.32	3985.47	3721.83	8.04	8.55	6.58	9.47
8	693.67	4005.12	3736.55	7.95	8.60	6.04	8.71
9	901.55	4020.24	3785.18	8.02	8.49	5.79	8.25
10	1157.42	4043.12	3772.26	7.98	8.32	5.30	7.98

number of moves for each example set, the average value of the variable cost obtained from each heuristic over the 100 examples of the set ($\bar{Z}_g^v$ for *GREEDY* and $\bar{Z}_b^v$ for *ABP*), the average number of transporters obtained by the heuristics ($\bar{R}_g$ for *GREEDY* and $\bar{R}_b$ for *ABP*), and the average percentage of time the transporters are idle within the time period T:

$$\bar{lc}_{g,b} = \frac{\sum_{k \in V} Ty_k - Z_{g,b}}{\sum_{k \in V} Ty_k} \times 100\%$$

Idle time indicates poor allocation of fixed cost. The subscripts g and b refer to the solutions derived by *GREEDY* and *ABP*, respectively.

Interesting trends are depicted in Table 1. The average variable cost as derived by the heuristics remains nearly constant; thus, the unloaded traveling time remains nearly constant as well. Furthermore, the average number of transporters decreases, as the number of moves increases and the inter-resource distances decrease. This is expected, since the packing of smaller moves to transporters is easier and more efficient for both heuristics; in addition, the return moves that close the loop for each transporter become smaller. Comparing the results for the two heuristics, it is clear that ABP outperforms $GREEDY$ with respect to the variable cost. The opposite occurs in terms of activated transporters. From the last two columns of Table 1, it is evident that as the average length of transporter moves decreases, the idle time also decreases. This is due to better assignment of complete moves to transporters. Finally, the transporters obtained by $GREEDY$ are better utilized than those derived by ABP.

In Table 2 the performance of the heuristics with respect to the deviation from the assignment lower bounds is summarized. Column 2 lists the average deviation of the solution derived by $GREEDY$ from the lower bound Z_a^* of the variable cost. Column 3 lists the average deviation of the number of transporters derived by $GREEDY$ from the lower bound R_a^*. Columns 4 and 5 list the same measures for the ABP heuristic.

Table 2: Performance of the heuristics: solutions vs. lower bounds

Example	*Heuristic GREEDY*		*Heuristic ABP*	
set number	$\overline{\frac{Z_g^v - Z_a^*}{Z_a^*}} \times 100$ %	$\overline{\frac{R_g - R_a^*}{R_a^*}} \times 100$ (%)	$\overline{\frac{Z_b^v - Z_a^*}{Z_a^*}} \times 100$ %	$\overline{\frac{R_b - R_a^*}{R_a^*}} \times 100$ %
1	14.32	36.42	9.85	41.27
2	14.03	31.72	9.13	35.37
3	13.95	25.74	8.74	30.18
4	13.48	20.86	8.12	24.32
5	11.97	17.38	7.31	20.71
6	11.08	14.63	6.25	19.38
7	10.76	9.54	6.14	14.20
8	10.12	6.97	5.67	12.15
9	9.75	3.76	5.28	8.56
10	9.26	1.98	4.56	3.74

From Table 2 it is clear that as the problem size increases, while the total loaded traveling time remains approximately invariant, the heuristics perform better and the solution values approach those of the lower bounds. Furthermore, it is obvious that ABP is consistently closer to the assignment lower bound (variable cost), while the $GREEDY$ solutions are closer to the lower bound of the number of transporters.

It is emphasized that in manufacturing applications a large number of moves is typical, with small times associated to each one with respect to the time period T. For example, the shop of a radar assembly manufacturer in Baltimore comprises ~100 workstations, and produces about 5000 different parts with an average demand of 40 units per year. Since the average number of operations in the parts' routings is 3, the total number of inter-resource moves per year is about 600,000. Consequently, for a 300 days per year working schedule with a single shift per day, about 2000 moves per day should be performed by the transporters. Thus, both heuristic algorithms are expected to provide satisfactory results when applied to evaluate the number of necessary transporters and the optimal assignment of unloaded moves between manufacturing resources.

7 Conclusions

In this paper we have studied the problem of designing a material handling system that employs the minimum number of transporters to transfer material within a manufacturing facility with minimal handling effort. Fixed acquisition and variable operational costs were explicitly considered. An integer program was formulated to capture the trade-off between these two costs. To solve the resulting NP-hard optimization problem, we developed two heuristic solution approaches: the first allots in a greedy fashion moves to transporters, while the second starts from the optimal solution of the assignment problem and, after grouping moves to routes, allots them to transporters through a two-stage bin-packing procedure. The heuristics were analyzed in terms of computational time and worst-case performance, and extensive computational tests were executed to evaluate their average performance.

The computational results indicate that both heuristics are efficient, and adequate to support the material handling system design process. Their performance drastically improves when the size of the problems, in terms of the total number of inter-resource moves, increases. $GREEDY$ solutions diverge from the lower bound of the number of required transporters (provided by the assignment problem $\mathcal{A}$) less than 2% in large-size problem instances; thus, if the fixed cost is the primary consideration, $GREEDY$ seems to be more appropriate. On the other hand, ABP solutions diverge from the variable cost lower bound less than 5% for large-size problem instances; consequently, this heuristic is more appropriate when material handling cost is the main consideration.

The mathematical formulation accurately models the design-level control problem if the design horizon T is a relatively small period (e.g. a shift). In such cases the same pattern of material flow is repeated in each period and the unloaded travel of transporters during shop operation is expected to be relatively close to that resulting from the solution pf $\mathcal{P}$. However, if the manufacturing system is not expected to demonstrate consistent periodicity, the model may not capture the effects of on-line control. In this case, the assumption of availability of inventory to be transfered at each resource may not be satisfied. The may, in turn, lead to an underestimation of the unloaded travel and the number of transporters required. To overcome this drawback, a time window [5] may be introduced to reflect the time within which each loaded move is to be performed. To incorporate time windows in the formulation, minor modifications in the graph of moves are necessary to reflect feasible move sequences [5].

In addition to offering a good initial estimate of the number of transporters required for a manufacturing system, the algorithms presented here may be integrated with facility design methods. Given a machine layout, the material handling flow paths can be optimally designed to provide the actual inter-resource distances; subsequently, the number of transporters and the unloaded moves can be evaluated by solving P. This will provide a realistic cost for the shop layout under consideration. The procedure can be repeated by generating new layouts, until a better global solution, in terms of total material handling investment and operational costs, is found. Search techniques such as simulated annealing or genetic algorithms could be employed in such an integrated facility design methodology.

Acknowledgments

This work was supported in part by the National Science Foundation under Grants # NSF D CDR 8803012 and NSF EEC 94-02384. We would like to thank Michael J. Vukovich for programming the Hungarian and FFD algorithms on the CIM Lab Sun Sparc Workstations.

Appendix A

This Appendix provides the proofs of the theorems of Section 5.

Proof of Theorem 1

At each iteration of $GREEDY$ exactly one loaded move is assigned to a transporter. However, when the capacity of a transporter is exceeded the last move of the current iteration must be reexamined in a subsequent iteration, since it cannot be assigned to the current transporter. This will occur at most once for each transporter, and if I is the total number of iterations, we obtain: $I \leq n + \sum_{k \in V} y_k$. From inequality (10), $\sum_{k \in V} y_k < \lceil n/2 \rceil$. Thus, $I \leq n + \lceil \frac{n}{2} \rceil \leq \frac{3n}{2} + 1$.

Identifying the minimum row element of the matrix $[c_{ij}]$ in Step 5 of $GREEDY$, requires at most n comparisons at each iteration; actually, we need $n-1$ at the first iteration, $n-2$ at the second one, etc. As a result, the maximum total number of operations, OP, of this heuristic is bounded as follows:

$$OP < n \cdot I \leq n \cdot (\frac{3n}{2} + 1) = \frac{3n^2 + 2n}{2} \tag{16}$$

Consequently, the computational complexity of $GREEDY$ is $O(n^2)$.

Proof of Theorem 2

The primal-dual algorithm for the assignment problem in Step 1 of ABP requires $O(n^3)$ operations, since the n loaded moves in L must be matched [15]. The identification of routes and route sets in Steps 4 through 9 requires inspection of each loaded move in the sequences given by the optimal solution of $\mathcal{A}$. Thus, exactly n operations are required before the criterion of Step 9 is no longer satisfied.

Since the maximum number of routes is $n/2$, Step 10 would require at most $(n/2) \cdot log(n/2)$ operations to sort them in decreasing order of their lengths [4] and $n/2$ operations to pack these routes into transporters of size T [11]. Finally, the enumeration of the best connecting points in Step 14 requires at most $O(n^2)$ operations, as shown below. Cosider K transporters to be merged, and let $n_1, ..., n_K$ be the number of moves assigned to each of them. Then in the worst case, Step 14 would examine each possible pair. This would require $\sum_{i=1}^{K} \sum_{j=1, j \neq i}^{K} n_i n_j$ operations. Since $\sum_{i=1}^{K} n_i = n$, the maximum number of operations at this Step is n^2.

Combining the results of above arguments, the total number of operations, OP, is bounded by:

$$OP \leq n^3 + n + \frac{n}{2} \cdot log(\frac{n}{2}) + \frac{n}{2} + n^2 \tag{17}$$

Consequently, the computational complexity of ABP is $O(n^3)$.

Proof of Theorem 3

The *GREEDY* algorithm selects the minimum element of the cost matrix in the first iteration. However, in the remaining $n-1$ iterations the algorithm may select elements different than those of the optimal solution. Since the maximum deviation of the cost elements is $c_{max} - c_{min}$, it follows that:

$$Z_g^v - Z_{opt}^v \leq (n-1) \cdot (c_{max} - c_{min}) < n \cdot (c_{max} - c_{min}) \tag{18}$$

The variable cost of the optimal solution to problem $\mathcal{P}$ is bounded from below by nc_{min}: Thus, from inequality (18), we conclude that:

$$\frac{Z_g^v - Z_{opt}^v}{Z_{opt}^v} < \frac{n(c_{max} - c_{min})}{nc_{min}} \quad \Leftrightarrow \quad \frac{Z_g^v}{Z_{opt}^v} < \frac{c_{max}}{c_{min}}$$

Lemma 1 shows that the bound provided by the first of inequalities (18) is tight, i.e. there exist problem instances of $\mathcal{P}$, for which (18) holds at equality.

Lemma 1 *There exists an $n \times n$ matrix $[c_{ij}]$ of cost coefficients, a time horizon T, and a sequence $Q = \{p_1, p_2, ...\}$ of loaded moves selected in Step 2 of GREEDY for which:*

$$Z_g^v - Z_{opt}^v = (n-1) \cdot (c_{max} - c_{min})$$

Proof Consider an instance of problem $\mathcal{P}$ with $c_{max} = 2c_{min}$ and $T = 5c_{min}$. Let $[c_{ij}]$ be the 5×5 matrix with entries c_{max} and c_{min} only, as shown below:

$$[c_{ij}] = \begin{bmatrix} \infty & c_{min} & c_{min} & c_{min} & c_{min} \\ c_{min} & \infty & c_{max} & c_{max} & c_{max} \\ c_{max} & c_{min} & \infty & c_{min} & c_{min} \\ c_{min} & c_{min} & c_{min} & \infty & c_{max} \\ c_{min} & c_{min} & c_{min} & c_{max} & \infty \end{bmatrix}$$

We claim that for this problem instance (18) holds at equality.

Let $Q = \{1, 4\}$ be a sequence that could be followed by *GREEDY*, which could provide the following matchings (assigned to the two transporters, l_1 and l_2), depending on the tie-breaking rules:

$$l_1 = \{(1,2), (2,3), (3,1)\} \quad \text{subtour length} = c_{min} + 2c_{max} = T$$
$$l_2 = \{(4,5), (5,4)\} \quad \text{subtour length} = 2c_{max} < T$$

The resulting variable cost is $Z_g^v = c_{min} + 4c_{max}$. The optimal solution to this instance of $\mathcal{P}$ gives the following assignment matchings for a single transporter: $l_1 = \{(1,5), (5,3), (3,4), (4,2), (2,1)\}$, and the variable cost is $Z_{opt}^v = 5c_{min}$. Thus, $Z_g^v - Z_{opt}^v = 4(c_{max} - c_{min})$.

Proof of Theorem 4

Consider the subtours derived by solving the assignment problem corresponding to $\mathcal{P}$. If each subtour comprises two moves of length c_{min}, and since the maximum number of subtours is $n/2$, the total variable cost is $Z_a^* = n \cdot c_{min}$. When ABP attempts to connect route sets, the total cost is augmented at most by $2 \cdot (c_{max} - c_{min})$. This connection removes two complete moves from the solution and introduces two new complete moves. In the worst case, $\lceil n/2 \rceil - 1$ connections will be performed, and, consequently

$$Z_b^v - Z_a^* \leq (\lceil \frac{n}{2} \rceil - 1) \cdot (2c_{max} - 2c_{min}) \leq n \cdot (c_{max} - c_{min}) \tag{19}$$

From (19) and the assignment cost, we conclude that:

$$\frac{Z_b^v - Z_a^*}{Z_a^*} \leq \frac{(c_{max} - c_{min})}{c_{min}} \qquad \Leftrightarrow \qquad \frac{Z_b^v}{Z_a^*} \leq \frac{c_{max}}{c_{min}}$$

Proof of Theorem 5

The optimal number of transporters is bounded from below by the ratio of the variable cost over the design horizon T, i.e., $R_{opt} \geq \lceil \frac{Z_{opt}^v}{T} \rceil \geq \frac{Z_{opt}^v}{T}$. From inequality (10) we know that $R_h \leq n/2$. Also $Z_{opt}^v \geq nc_{min}$, and consequently $R_h \leq \frac{Z_{opt}^v}{2c_{min}}$. Thus,

$$\frac{R_h}{R_{opt}} \leq \frac{1}{2} \cdot \frac{T}{c_{min}}$$

References

[1] Bartholdi, J.J.III and Platzmann, L.K., 1988. Decentralized control of a fixed route automated guided vehicle system, *IIE Transactions*, vol. 21, no. 1, pp. 76-81.

[2] Bozer, Y.A. and Srinivasan, M.M., 1991. Tandem configurations for automated guided vehicle systems and the analysis of single vehicle loops, *IIE Transactions*, vol. 23, no. 1, pp. 23-27.

[3] Bozer, Y.A. and Srinivasan, M.M., 1992. Tandem AGV systems: a partitioning algorithm and performance comparison with conventional AGV systems, *European Journal of Operational Research*, vol. 63, no. 2, pp. 173-191.

[4] Cormen, T.H., Leiserson, C.E. and Rivest, R.L., 1990. *Introduction to algorithms*, MacGraw-Hill Publications, New York.

[5] Desrosiers, J., Dumas, Y., Solomon, M.M., and Soumis, F., 1994. Time constrained routing and scheduling, in M.E. Ball, T.L. Magnanti, C. Monma, and G.L. Nemhauser (eds.), *Handbook in Operations Research and Management Science*, volume on Networks, to appear.

[6] Egbelu, P.J., 1987. The use of non-simulation approaches in estimating vehicle requirements in an automated guided vehicle based transport system, *Material Flow*, vol.4, pp. 17-32.

[7] Golden, B.L. and Assad, A.A., 1988. *Vehicle routing: methods and studies*, Elsevier Science Publishers, Amsterdam.

[8] Heragu, S.S., 1992. Recent models and techniques for solving the layout problem, *European Journal of Operational Research*, vol. 57, no. 2, pp. 1-13.

[9] Ioannou, G., Minis, I., Nagi, R. and Proth, J.M., 1994. Design of material handling flow paths, *Proceedings of the 1994 NSF Design & Manufacturing Grantees Conference*, Cambridge, MA, pp. 333-334.

[10] Jacobs-Blecha, C. and Goetschalckx, M., 1992. Analysis of horizontal transportation systems with multiload carriers, *Proceedings of the 1992 NSF Design and Manufacturing Systems Conference*, Atlanta, GA, pp. 871-874.

[11] Johnson, D., Demers, A., Ullman, J., Garey, M. and Graham, R., 1974. Worst-case performance bounds for simple one-dimensional packing algorithms, *SIAM Journal of Computing*, vol. 3, pp. 297-325.

[12] Lewis, R.T. and Parker, R.G., 1982. On a generalized bin-packing problem, *Naval Research Logistics Quarterly*, vol. 29, no. 1, pp. 119-145.

[13] Maxwell, W.L. and Muckstadt, J.A., 1982. Design of automated guided vehicle systems, *IIE Transactions*, vol. 14, no. 2, pp. 114-124.

[14] Montreuil, B., Venkatardi, U. and Ratliff, H.D., 1993. Generating a layout from a design skeleton, *IIE Transactions*, vol. 25, no. 1, pp. 3-15.

[15] Papadimitriou, C.H. and Steiglitz, K., 1982. *Combinatorial optimization: algorithms and complexity*, Prentice-Hall, Inc., Englewood Cliffs, NJ, pp. 247-255.

[16] Sinriech, D. and Tanchoco, J.M.A., 1992. An economic model for determining AGV fleet size, *International Journal of Production Research*, vol. 30, no. 6, pp. 1255-1268.

[17] Sinriech, D. and Tanchoco, J.M.A., 1992. Impact of unloaded vehicle flow on performance of single loop AGV systems, *International Journal of Production Research*, vol. 30, no. 10, pp. 2237-2252.

[18] Tanchoco, J.M.A., Egbelu, P.J. and Taghaboni, F., 1987. Determination of the total number of vehicles in an AGV-based material transport system, *Material Flow*, vol.4, pp. 33-51.

[19] Wysk, R.A., Egbelu, P.J., Zhou, C. and Ghosh, B.K., 1987. Use of spread sheet analysis for evaluating AGV systems, *Material Flow*, vol.4, pp. 53-64.