Journal of Industrial and Systems Engineering
Vol. 4, No. 3, pp 167-182
Fall 2010

A Genetic Based Scheduling Algorithm for the PHSP with Unequal Batch Size Inbound Trailers

Douglas L. McWilliams

Mississippi State University-Meridian, 1000 Highway 19 North, Meridian, MS 39307-5799
dmcwilliams@meridian.msstate.edu

ABSTRACT

This paper considers the parcel hub scheduling problem (PHSP) with unequal batch size inbound trailers, which is a combinatorial optimization problem commonly found in a parcel consolidation terminal in the parcel delivery industry (PDI). The problem consists of processing a large number of inbound trailers at a much smaller number of unload docks. The parcels in the inbound trailers must be unloaded, sorted and transferred to the load docks, and loaded onto the outbound trailers. Because the transfer operation is labor intensive and the PDI operates in a time-sensitive environment, the unloading, sorting, transferring, and loading of the parcels must be done in such a way as to minimize the timespan of the transfer operation. A genetic algorithm is used to solve the PHSP. An experimental analysis shows that the algorithm is able to produce solution results that are within 17% of the lower bound, 16% better than a competing heuristic, and 24% better than random scheduling.

Keyword: Distribution, Cross dock, Sortation, Genetic algorithm, Work load balancing received

1. INTRODUCTION

The parcel hub scheduling problem (PHSP) is a combinatorial optimization problem that involves assigning a large number of inbound trailers to a much smaller number of unload docks. The parcels in the inbound trailers must be unloaded and sorted across a network of conveyors to the outbound trailers at the load docks. The transfer of the parcels must be done in such a way as to minimize the timespan of the transfer operation and thus the operational cost of the transfer operation (Gue, 1996; Bartholdi and Gue, 2000). The assignment of inbound trailers to the unload docks and the sequence of the inbound trailers to the unload docks impact the amount of congestion in the transfer system and thus the timespan of the transfer operation.

The PHSP is common in the parcel delivery industry (PDI). Companies operating in the PDI include the United States Postal Service (USPS), United Parcel Service (UPS), Federal Express (FedEx), and Deutsche Post (DHL). In 2002, according to the Bureau of Transportation Statistics' Commodity Flow Survey, the PDI moved approximately 12 percent (more than US$1 trillion) of goods in the United States (Ammah-Tagoe (2006)). Hence, it is imperative that companies

---

ISSN: 1735-8272,

operating in the PDI continue to enhance performance to deliver goods to consumers fast and at least cost.

Figure 1 depicts a typical central parcel consolidation terminal (CPCT), which tends to have from 10 to 50 unload docks and from 50 to 160 load docks. Depending on the size of the CPCT, from 100 to 500 inbound trailers daily collectively containing from 60,000 to 300,000 parcels must be processed through a terminal.

A complex conveyor network is the primary transportation mode for moving parcels through the terminal. In Figure 1, the CPCT contains one input station which consists of three unload docks—Nodes U1, U2, and U3—with a primary sorter at each unload dock—Nodes P1, P2, and P3. There are three output stations—Nodes S1, S2, and S3. Each output station has a secondary sorter that diverts the parcels to the load docks—Nodes L1 through L9—for loading onto the outbound trailers; a unique destination point is assigned to each load dock (Masel and Goldsmith, 1997; Masel, 1998).

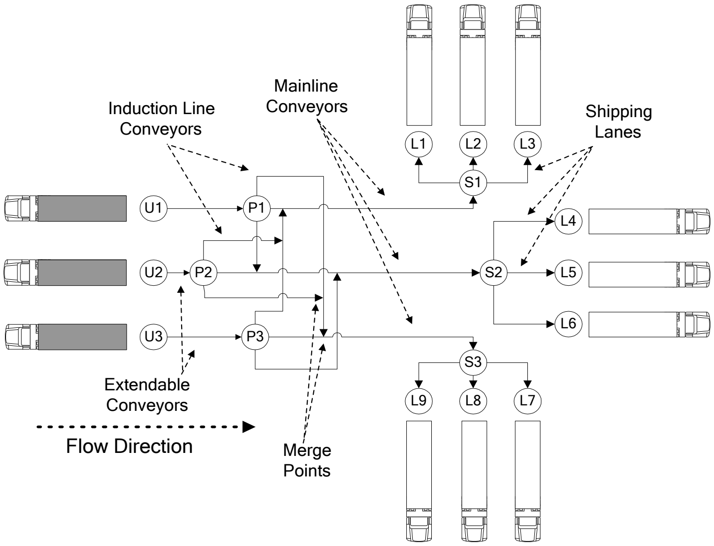


Figure 1 A Basic conveyor network

The conveyor network is made of several induction line conveyors, fewer mainline conveyors, and shipping lanes. At each unload dock, the primary sorters scan the address label on each parcel as the parcel exits the inbound trailer and divert the parcel onto the appropriate induction line conveyor. The induction line conveyors merge the parcels onto the downstream mainline conveyors that feed the output stations. There is typically one mainline conveyor per output station that feeds parcels to the secondary sorters (Gilbert, 1934; Goodison, 1981; Gould, 1991). The sorters may be manual laborers or computerized automated systems.

After outbound trailers are filled, the trailers are replaced by empty or partially filled outbound trailers for the same destination. The filled outbound trailers and some partially filled outbound trailers are eventually released to deliver the freight to the next destination terminal in the delivery network.

In this study, a genetic-based scheduling algorithm is proposed to solve the PHSP with unequal batch inbound trailers (UGBSA). The contribution of the research is an enhanced decision support tool to enable companies in the PDI to improve the productivity of transfer operations in CPCTs. The remainder of this paper is organized as follows. Section 2 provides a discussion of the relevant literature. The problem statement is presented in Section 3. In Section 4, an alternative approach to the timespan minimization is presented. A lower bound on the parcel workload objective function is presented in Section 5. A genetic-based scheduling heuristic presented in Section 6. Section 7 contains the computational study, and the summary and future research is presented in Section 8.

2. LITERATURE REVIEW

Only a small number of researchers have focused their attention on improving the operational performance of a CPCT. Rohrer (1995) discussed the importance of modeling a CPCT. Masel and Goldsmith (1997) and Masel (1998) proposed procedures to assign destinations to load docks. In this study, the parcel workload to load docks was balanced at the static level, using historical parcel data. For a real-world CPCT, the parcel mix entering the facility varies from day to day and during the transfer operation. As a result, parcel feed rates over the various paths from unload docks to load docks vary.

With variations in the parcel flow rates, some unload docks may have high feed rates, while others have very low feed rates or starved altogether. For instance, if the feed rate over a path to an output station (i.e., a set of load docks) exceeds the throughput capacity of the path, parcel flow on the path becomes congested. If the congestion persists for an extended time, the feed rate over the path slows or stalls preventing the flow of parcels to other loads docks in the output station. If the blockage propagates upstream to the input stations (i.e., sets of unload docks), the flow of parcels over the other paths becomes blocked, possibly preventing the flow of parcels through the entire facility. This congestion ultimately increases the timespan of the transfer operation and thus the operating cost. Hence, the timespan of a transfer operation is a summation of active flowtime (unload, load, and move time) through the terminal and passive flowtime (delay time) due to flow congestion. The objective is to minimize the passive flowtime.

McWilliams, Stanfield, and Geiger (2005) presented a simulation-based scheduling algorithm for the equal batch size inbound trailer problem (ESBSA) to generate unload schedules for the inbound trailers to minimize the passive flowtime and thus the timespan. The results showed reductions in the timespan of transfer operations from 10% to 25% compared to random scheduling. For a small CPCT, the results showed a timespan only 5% greater than a proposed lower bound for some problems. McWilliams et al. (2008) proposed an improved SBSA with a list-scheduling procedure that considered inbound trailers with unequal batch sizes. The results of the USBSA showed between a 2% and 5% incremental reduction in the timespan of the transfer operation.

The major drawback of the simulation-based scheduling algorithms was the dependency on simulation evaluations, which make obtaining solutions computationally expensive. The average computational time required to generate acceptable unload schedules for small-, medium-, and large-size problem sets were 20.6, 67.7, and 360.5 minutes, respectively. Such computational requirements make simulation-based scheduling impractical for real-world applications.

To minimize the computational time requirement, Lai (2007) proposed a surrogate search method that combined minimum simulation evaluations with a regression model to predict system behavior. Instead of focusing on the parcel flow pattern resulting from the unload schedule for inbound trailers, Lai chose to minimize passive flowtime by reacting to parcel flow variation, moving team of loaders to areas of the facility being impacted by high parcel flow rates. Lai explains that the weakness of this approach is that it does not take into consideration the stochastic nature of the PHSP.

McWilliams (2009) proposed a genetic-based scheduling algorithm (EGBSA) to solve the PHSP with equal batch size inbound trailers. McWilliams chose to minimize the passive flowtime by balancing the parcel workload through the terminal. A time-based decomposition approach was used. Time-based decomposition involves dividing the time horizon into several smaller sub-periods. Many researchers have used time-based decomposition to solve complex scheduling problems to near optimality (Basset, Pekny, and Reklaitis, 1966; Khmelnitsky, Kogan, and Maimon, 2000).

For the PHSP, each unload dock contained time buckets in which to allocate a single inbound trailer. The number of time buckets was a function of the number of inbound trailers and the number of unload docks. The EGBSA produced solutions with timespans that were from 10% to 23% lower than the solutions proposed in McWilliams et al. (2005, 2008). The average computational time required to generate unload schedules for small-, medium-, and large-size problem sets were respectively 0.58, 5.24, and 19.53 minutes compared to the respective 20.6, 67.7, and 360.5 minutes for the SBSA.

A major weakness of the EGBSA was the assumption that inbound trailers contain equal numbers of parcels (equal batch sizes). In this study, a genetic-based scheduling algorithm approach is developed and illustrated to solve the PHSP with unequal batch size inbound trailers (UGBSA).

3. PROBLEM STATEMENT

A CPCT has a set of inbound trailers. Each trailer contains a random number of parcels that are bound to various destination points in the delivery network. The inbound trailers must be processed at the unload docks, where the number of inbound trailers is much greater than the number of unload docks. A large number of parcels are required to be unloaded, sorted and transferred through the terminal directly to a set of load docks to be loaded onto the outbound trailers. A complex conveyor network is used to transport the parcels from the unload docks to the load docks (Figure 1). The route of each parcel is fully defined by the known unload docks and load docks and the configuration of the conveyor system.

The system is susceptible to excessive congestion which increases the timespan of a transfer operation and therefore increases the operational cost and decreases the delivery reliability of the delivery system. The objective is to find an unload schedule or set of schedules that minimizes the congestion in the system and thus the timespan of the transfer operation.

An unload schedule is defined as the allocation of inbound trailers to the unload docks and the sequence of the inbound trailers to the unload docks. The transfer time starts when the first inbound trailer begins processing at an unload dock and ends when the last parcel is loaded onto an outbound trailer at a load dock.

The following simplifying assumptions are implemented from McWilliams et al. (2008) and McWilliams (2009).

1. Unload docks are identical and unloading is a manual operation. An unload dock can process any trailer and the unload docks have equal and constant service rates;

2. Load docks are identical and loading is a manual operation. A load dock can process any trailer and the load docks have equal and constant service rates;
3. Parcels are transported from unload docks to load docks by means of a fixed network of conveyors;
4. All trailers both inbound and outbound are available at the beginning of the transfer operation;
5. Empty inbound trailers are instantaneously replaced with full inbound trailers;
6. Full outbound trailers are instantaneously replaced with empty outbound trailers;
7. All inbound and outbound trailers have equal priority; and
8. No trailer can be preempted once its unload or load processing has begun.

4. TIME-BASED DECOMPOSITION

McWilliams (2009) presented an EGBSA with time-based decomposition to solve the PHSP with equal batch size inbound trailers wherein the time horizon at the unload docks are partitioned into time buckets. Each time bucket has the capacity to process a single inbound trailer. A similar solution approach is sought to solve the PHSP with unequal batch size inbound trailers. The main difference is that each time bucket now has the capacity to process a predetermined number of parcels. This modification gives the algorithm the needed flexibility to produce quality unload schedules for unequal batch size inbound trailers.

The following notation is used in the development of the UGBSA:

U	set of unload docks
S	set of secondary sorters
B	set of time buckets
T	set of inbound trailers
a_i	number of parcels in inbound trailer i
h_{is}	number of parcels in inbound trailer i that must pass through secondary sorter s
v_i	number of time buckets required to process inbound trailer i
D_i	set of parcel segments in inbound trailer i (a function of v_i)
E_j	set of inbound trailers assigned to time bucket j (an inbound trailer can be assigned to one or more time buckets based on D_i)
c	number of parcels that can be unloaded from an inbound trailer at each unload dock per time bucket
A_u	cumulative number of parcels assigned to unload dock u
i	index value for inbound trailers
k	index value for the inbound trailer segments
s	index value for the secondary sorters
u	index value for the unload docks
j	index value for the time buckets

The list of inbound trailers in Table 1 is used in the illustration of the proposed solution approach. Inbound trailer 1 (T1) contains two parcels that must be routed to the secondary sorter at output station 1, one parcel that must be routed to the secondary sorter at output station 2, and two parcels that must be routed to the secondary sorter at output station 3. Inbound trailer 2 (T2) contains six parcels that must be routed to the secondary sorter at output station 1, nine parcels that must be routed to the secondary sorter at output station 2, zero parcels that must be routed to the secondary sorter at output station 3, and so forth. The number of parcels in each inbound trailer is 5, 10, or 15. The number of time buckets (TB) required to process each inbound trailer is

$$v_i = \frac{a_i}{c} \qquad \forall i \in T \tag{1}$$

Table 1 Unequal batch size inbound trailers

T_i	h_{i1}	h_{i2}	h_{i3}
1	2	1	2
2	6	9	0
3	4	4	2
4	1	4	0
5	0	6	4
6	9	3	3
7	2	0	8
8	2	1	2
9	2	2	1
10	2	0	8

For the illustration, *c* is equal to 5 parcels per time bucket at each unload dock. Hence, the number of time buckets required to unload the parcels from an inbound trailer is 1, 2, or 3 (Table 2). As a result, the number of parcel segments in inbound trailer *i* is v_i ($|D_i| = v_i$).

Table 2 Required number of time buckets

Inbound Trailer									
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
1	3	2	1	2	3	2	1	1	2

For inbound trailers that require more than one time bucket, the simplifying assumption is that the parcel types are uniformly distributed over the time buckets, which may or may not be the actual case for any given inbound trailer. For instance, because inbound trailer 2 requires three time buckets, the assumption is that the inbound trailer is divided into three equal segments and that each segment must be process successively. In each segment, two parcels are bound for the secondary sorter at output station 1; three parcels are bound for the secondary sorter at output station 2, and zero parcels are bound for the secondary sorter at output station 3. Because inbound trailer 3 requires two time buckets, the assumption is that this inbound trailer is divided into two equal segments and that each segment must be process successively. In each segment, two parcels are bound for the secondary sorter at output station 1; two parcels are bound for the secondary sorter at output station 2, and one parcel is bound for the secondary sorter at output station 3.

Figure 2 shows the parcel workload profile at the unload docks for a random unload schedule. Inbound trailers 10, 1, and 3 are assigned to unload dock 1 in that order; inbound trailers 8, 4, 5, and 7 are assigned to unload dock 2 in that order, and inbound trailers 2, 9, and 6 are assigned to unload dock 3 in that order. The rectangles at the unload docks represent the assigned inbound trailers to the unload docks in the time buckets. The rectangles are partitioned to show the number of parcel types in inbound trailer *i* segment *k* that is bound to secondary sorter *s* in time bucket *j*. For instance, inbound trailer 10 is partitioned into two segments; inbound trailer 1 is partitioned into one segment, inbound trailer 3 is partitioned into two segments, and so forth. Hence, five time buckets are required to process the inbound trailers at unload dock 1, 6 time buckets at unload dock 2, and 7 time buckets at unload dock 3. Because the placement of parcel types in each inbound trailer is unknown, the rectangles do not indicate the location of the parcel types in the inbound trailers.

Figure 3 shows the corresponding parcel workload profile for the secondary sorters at the output stations over the time buckets. Let LD_{sj} denote the number of parcels that must be routed to secondary sorter *s* in time bucket *j* based on the inbound trailer assignment. Then,

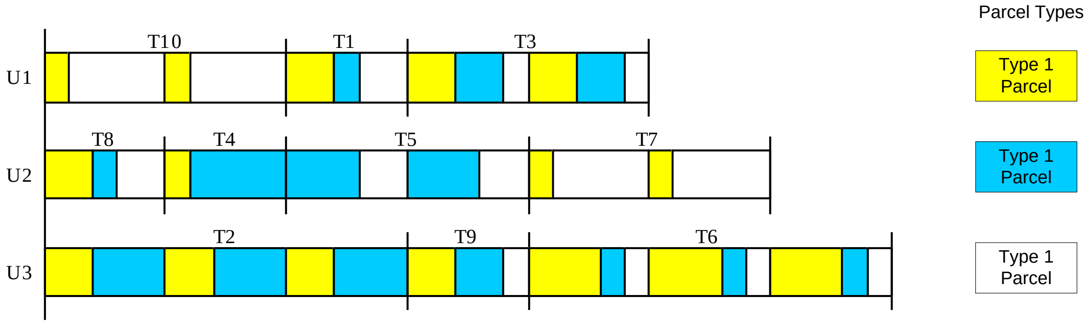


Figure 2 Unload docks parcel load profile based on unload schedule

$$LD_{sj} = \sum_{i \in E_j} \frac{h_{is}}{v_i} \qquad \forall s \in S; j \in B. \tag{2}$$

The secondary sorter with the maximum parcel workload in time bucket *j* is

$$LD_j = \max_{s \in S} \left\{ LD_{sj} \right\} \qquad \forall j \in B. \tag{3}$$

The sum of the maximum parcel workloads over the time buckets is

$$LD = \sum_{j \in B} LD_j \tag{4}$$

Based on Equations (2-4), LD_1 is equal to 6; LD_2 is equal to 7; LD_3 is equal to 7; LD_4 is equal to 7; LD_5 is equal to 6; LD_6 is equal to 5, and LD_7 is equal to 3. Hence, LD is equal to 41.

To minimize the passive flowtime, the goal is to balance the parcel workload to the secondary sorters over the time buckets. The following 0-1 programming model is used to find the optimal unload schedule:

$$\text{Minimize } LD = \sum_{j \in B} LD_j \tag{5a}$$

Subject to

$$\sum_{i\in T}\sum_{k\in D_i} x_{ikj} \le |U| \qquad \forall j \in B \tag{5b}$$

$$\sum_{j\in B} x_{ikj} = 1 \qquad \forall i \in T; k \in D_i \tag{5c}$$

$$v_i . x_{i1j} - \sum_{k\in D} x_{ik(j+k-1)} = 0 \qquad \forall\, i \in T;\ j \in B \tag{5d}$$

$$\sum_{i\in T} \frac{h_{is}}{v_i} \cdot \sum_{k\in D_i} x_{ikj} \le LD_j \qquad \forall s \in S;\ j \in B \tag{5e}$$

$$x_{ikj} = 0 \text{ or } 1 \qquad \forall i \in T; k \in D_i;\ j \in B \tag{5f}$$

where x_{ikj} is the decision variable that is equal to 1 if inbound trailer i segment k is assigned to time bucket j and 0 otherwise. Constraint (5b) ensures that no more than $|U|$ inbound trailer segments are assigned to a single time bucket j since there are only $|U|$ unload docks. Constraint (5c) ensures that inbound trailer i segment k is assigned to only one time bucket. Constraint (5d) ensures that the inbound trailer segments of a single inbound trailer are successively processed based on the assignment of the first inbound trailer segment. Constraint (5e) determines the maximum workload over the output stations during time bucket j based the inbound trailer assignment to the time buckets. Finally, Constraint (5f) ensures that the decision variables take on binary values.

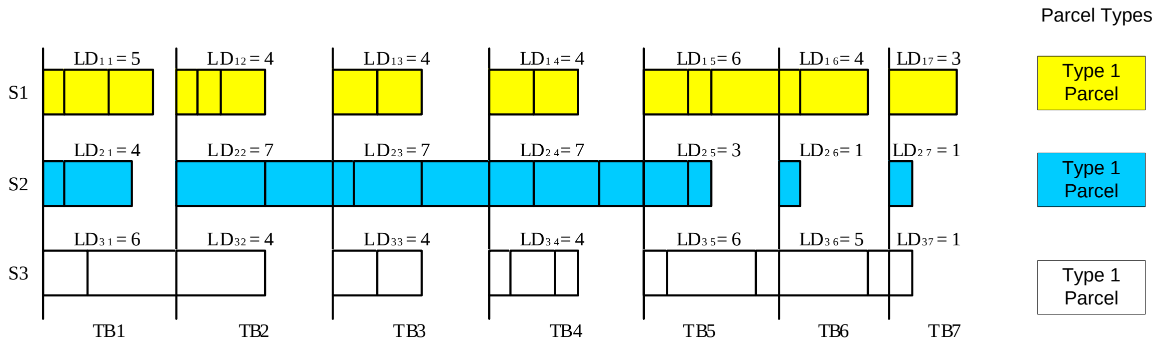


Figure 3 Output station parcel load profile based on unload schedule

Figures 4 and 5 show the parcel workload profile for the unload docks and the corresponding parcel workload profile to the secondary sorters resulting from the optimum unload schedule given the use of the 0-1 programming model. Inbound trailers 1, 8, 7 and 3 are assigned to unload dock 1 in that order; inbound trailers 6 and 2 are assigned to unload dock 2 in that order, and inbound trailers 5, 4, 9, and 10 are assigned to unload dock 3 in that order. The parcel workloads to the secondary sorters are equal to 5 for each time bucket. Because there are six time buckets, *LD* is equal to 30. This workload value (*LD* = 30) should be compared to the workload value (*LD* = 41) from the random schedule previously mentioned.

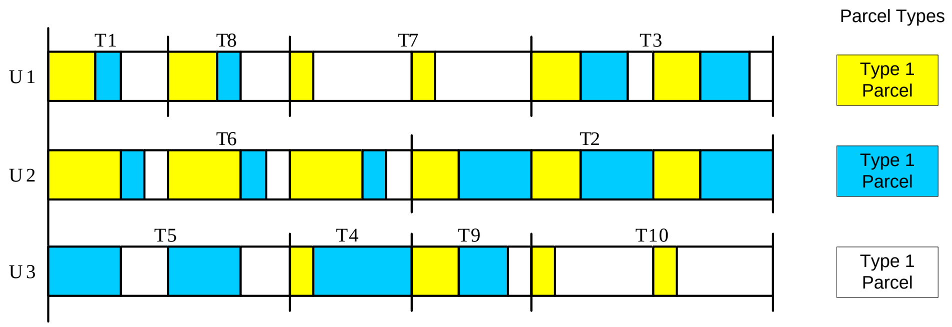


Figure 4 Revised unload docks parcel load profile based on inbound trailer assignments

5. LOWER BOUND

A lower bound is determined for the parcel workload based on McWilliams (2009). The assumption is that the parcel move times are negligible. Let LD^* denote the lower bound of LD. Then, for any given unload schedule, $LD \geq LD^*$. The lower bound can be computed from any arbitrary unload schedule:

$$LD^* = \frac{1}{s} \sum_{j \in B} \sum_{s \in S} LD_{sj} \cdot \qquad (6)$$

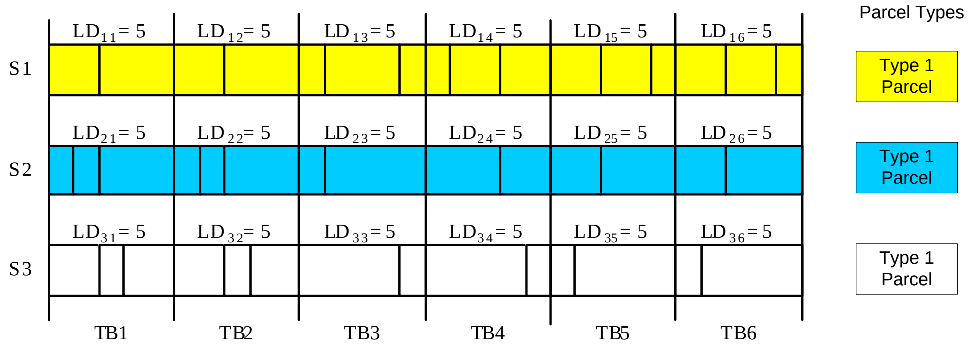


Figure 5 Revised output station parcel load profile based on inbound trailer assignments

6. GENETIC ALGORITHMS

Genetic algorithms (GAs) are popular techniques for addressing both constrained and unconstrained optimization problems, and they have been utilized in many application areas including, but not limited to, operations scheduling and sequencing, transportation scheduling, maintenance and reliability, financial portfolio management, and many others. GAs are stochastic search techniques that were inspired by the Darwinian theory of natural selection ("survival of the fittest") and use principles of biogenetics. GAs begin with a set of candidate solutions to a particular problem of interest. In the biological context, this set of solutions is considered a population of chromosomes, where each chromosome is a possible solution and is described by its genotype. Each gene in the chromosomal structure is the genotype's fundamental information element. All major operations of GAs are performed on the genotype level.

The population of chromosomes (solutions) evolves over successive generations by exchanging and randomly modifying genotype information of “parent” solutions in one generation creating “offspring” solutions in the next generation. Offspring solutions are created by merging two or more parent solutions using a recombination operation, and/or randomly modifying a solution with the use of a mutation operation. During each generation, the solutions are evaluated on their performance with respect to their ability to solve the problem of interest (i.e., their fitness is computed). The best fit solutions are assigned higher survival probabilities and are selected to serve as parents for reproduction to populate subsequent generations. The least fit solutions are given low survival probabilities and eventually die off. Evolution of the population of solutions continues until pre-specified termination conditions are met. The final set of solutions hopefully represents the optimal or near-optimal solutions to the problem (Goldberg, 1989; Haupt and Haupt, 1998).

Using the core components of a GA, the proposed solution approach for the PHSP with unequal batch sizes is as follows:

Step 1: Generate an initial population of permutations of randomly-sequenced inbound trailers;
Step 2: Construct unload schedules by assigning the inbound trailers to the unload docks based on the list-scheduling heuristic that is described later in Section 6.2;
Step 3: Compute the quality (fitness) of each unload schedule by determining the sum of the maximum parcel workloads (Equations 2-4);
Step 4: Perform the GA operations to obtain the next population of candidate solutions; and
Step 5: Repeat Steps 2-4 until the termination criteria are satisfied.

6.1. Solution representation

The fundamental decision of inbound trailer assignment is deciding at which unload dock should an inbound trailer be unloaded and in what order should the inbound trailers be unloaded at that dock. Using a genetic-based search approach to solve the PHSP requires the definition of the solution representation scheme. The representation scheme defines the window by which the search views the solution space, and, if not chosen carefully, can prevent the search from discovering effective solutions. For the PHSP, the solution representational scheme (genotype) is a permutation (or list) of numbers. Each number (gene) in the permutation is an identifier for an inbound trailer. For instance, suppose the CPCT in Figure 1 has ten inbound trailers identified as T1, T2, T3, …, T10 that must be processed through the facility, a possible permutation representation is T2-T8-T10-T4-T5-T1-T9-T3-T6-T7.

6.2. List-Scheduling heuristic

As with McWilliams et al. (2008) and McWilliams (2009), the assumption is that the unload docks are identical parallel resources (Assumption 1). As a result, the number of parcels in each inbound trailers directly impacts the parcel workload balance to the unload docks. A list-scheduling heuristic similar to the longest processing time (LPT) scheduling heuristic for the parallel machine scheduling problem is proposed to balance the parcel workload to the unload docks. The objective of the LPT scheduling heuristic is to minimize the maximum completion time of the jobs by balancing the workload to the machines (Graham, 1969). The heuristic for balancing the workload to the unload docks is as follows:

Step1: Set $A_u = 0$ for each unload dock u;
Step2: Take permutation w, which is a list of inbound trailers;
Step3: Select the first $|U|$ inbound trailers from the list;

Step4: Assign one inbound trailer to each unload dock;

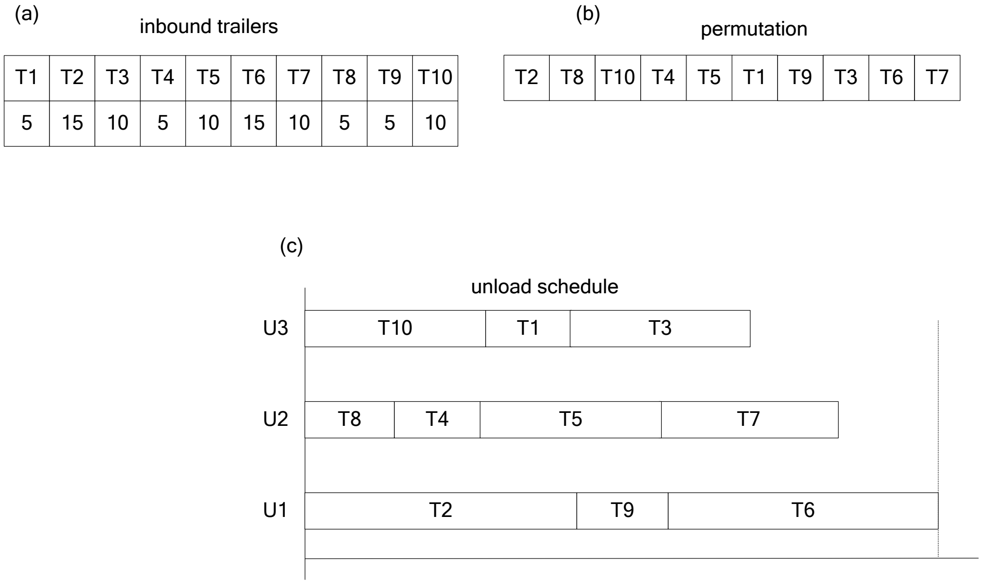
Step5: Update A_u $\left(A_u = A_u + a_{[i]}\right)$ for each unload dock *u*;

Step6: Select the next unassigned inbound trailer from permutation *w*;

Step7: Assign the inbound trailer to unload dock *u* with the smallest assigned parcel workload $\left(u' \in U \mid A_{u'} = \min_{u \in U}\left\{A_u\right\}\right)$; and

Step8: Repeat Steps 5 and 7 until all inbound trailers have been assigned to an unload dock.

Figure 6 is an illustration of the list-scheduling heuristic. The heuristic was used to produce the unload schedule in Figure 2. Figure 6a is a list of the inbound trailers with the number of parcels from Table 1. Figure 6b is a permutation of the inbound trailers generated by the GA. Figure 6c shows the unload schedule resulting from the list-scheduling procedure.



(a) inbound trailers

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
5	15	10	5	10	15	10	5	5	10

(b) permutation

T2	T8	T10	T4	T5	T1	T9	T3	T6	T7

Figure 6 A list-scheduling procedure

6.3. Genetic search operators – crossover and mutation

The crossover and mutation operators used in this study are the cycle crossover and two-position interchange operators, respectively. The logic for the cycle crossover operator is shown in Figure 7. The logic for the mutation operator is shown in Figure 8. Throughout the GA search process, fitness-proportionate selection is used for selecting the parent solutions for reproduction (Goldberg, 1989; Haupt and Haupt, 1998). Additionally, the elitist strategy is used to ensure that the best solution from the previous generation is carried over to the next generation.

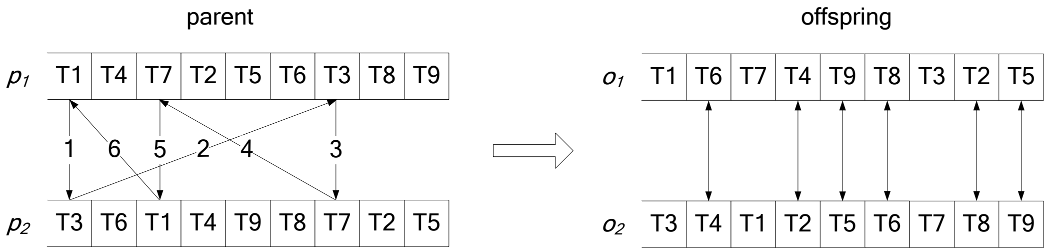


Figure 7 The cycle crossover operator

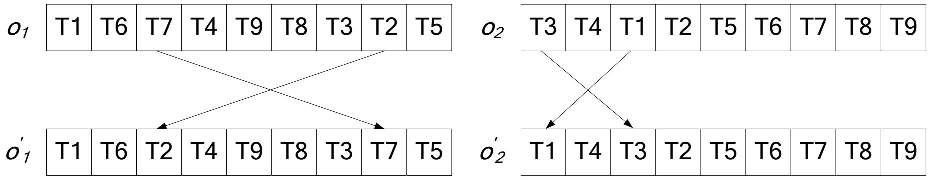


Figure 8 The two-position mutation operator

The general flow logic of the UGBSA is shown in Figure 9. The CPCT block contains the terminal configuration information, and the inbound trailer block contains information about the inbound trailers and their parcel content. The information is passed to the solution procedure. Within the solution procedure are the mechanics of the algorithm. The Start node begins the process of retrieving the required information and generating the initial population for the GA. The Selection block selects the candidate solution for reproduction, and the Reproduction block performs the crossover and mutation operations. The Evaluation block computes the fitness of the candidate solution. The List-Scheduling procedure selects one solution at a time and constructs the unload schedule from the permutation and passes the schedule to the fitness function for evaluation. When the termination criteria are met, the solution procedure yields the best solution found.

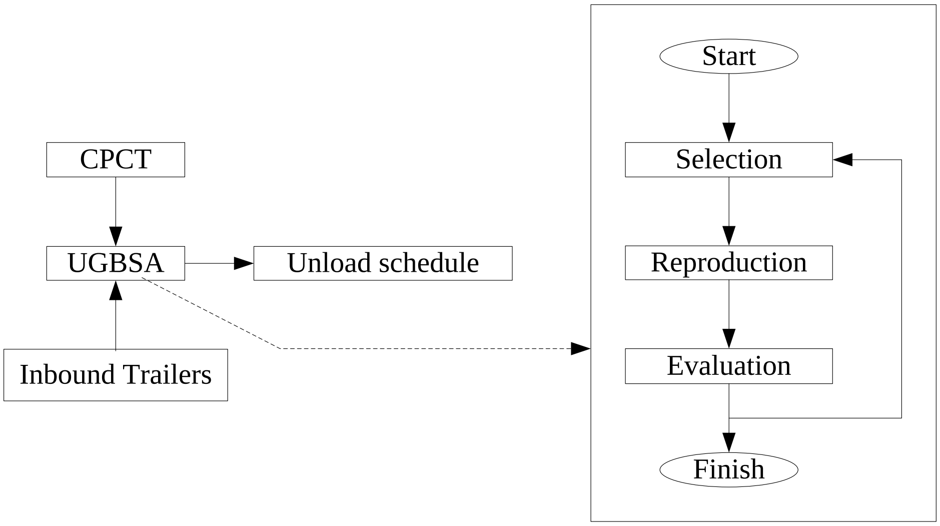


Figure 9 The operational architecture of the UGBSA

7. DESIGN IMPLICATION

7.1. Test problem generation

An experimental analysis is conducted to evaluate the UGBSA. The experimental conditions are created based on the experimental design of McWilliams et al. (2008). The throughput capacities of the resources and buffer capacity between resources are defined to reflect a real-world system. Three types of consolidation terminals are considered: a small terminal, a medium terminal, and a large terminal. The small terminal has three unload docks and nine load docks; there are three output stations each with three load docks. The medium terminal has 10 unload docks and 32 load docks; there are four output stations each with eight load docks. Lastly, the large terminal has 50 unload docks and 160 load docks; for this terminal, there are 20 output stations each with eight load docks.

The throughput capacities of each unload dock and each primary sorter is set at 600 parcels per hour. The throughput capacity of each final sorter is set at 600 parcels per hour for the small terminal and 1,500 parcels per hour for both the medium and large terminals. The throughput capacity of each load dock is set at 300 parcels per hour. As a result, the throughput capacities of the overall CPCTs are 1,800, 6,000, and 30,000 parcels per hour respectively for the small, medium, and large terminals.

A total of 300 test problems are generated. Each inbound trailer contains from 100 to 800 parcels. For the small terminal, 50 test problems each containing a total of 10,800 parcels and 50 test problem each containing a total of 18,000 parcels are generated. For the medium terminal, 50 test problems each containing a total of 36,000 parcels and 50 test problem each containing a total of 60,000 parcels are generated. Lastly, for the large terminal, 50 test problems each containing a total of 180,000 parcels and 50 test problem each containing a total of 300,000 parcels are generated. The experimental design for the CPCT is summarized in Table 3 with the maximum throughput capacity of the system (parcels per minute).

Table 3 Experimental design

Facility	No. Unload Docks	No. Load Docks	Throughput Capacity	No. Parcels
Small	3	9	30	10,800
				18,000
Medium	10	32	100	36,000
				60,000
Large	50	160	500	180,000
				300,000

7.2. Experimental implementation

The performance of the UGBSA is evaluated against the EGBSA from McWilliams (2009), random scheduling (RAND), and the lower bound (Equation 6). The solution quality of the algorithms can be revealed by taking the ratios of the various parcel workload values. Let Q denote the set of scheduling algorithms, and let $LD(q, c)$ denote the workload value resulting from scheduling algorithm q on problem c and $LD^m(c)$ denote the “minimum” obtained workload value over the scheduling algorithms on problem c. Then, the algorithm quality can be computed as

$$\beta(q) = \frac{\sum_{c \in C} LD(q,c)}{\sum_{c \in C} LD^m(c)} \qquad \forall q \in Q \tag{7}$$

Now let $LD^*(c)$ denote the lower bound for the parcel workload value on problem c. The algorithm quality with respect to the lower bound can be computed as

$$\eta(q) = \frac{\sum_{c \in C} LD(q,c)}{\sum_{c \in C} LD^*(c)} \qquad \forall q \in Q \tag{8}$$

Other performance criteria include the number of times scheduling algorithm q found the best solution ($\Psi(q)$) and the required computation time ($CPU(q)$).

The GA for the SBSA is coded using C Language. The simulation of the hub terminal is implemented using Arena 8.01 simulation software. The experiments are run on a computer workstation with a 1.3 GHz Pentium 4 processor, 256 MB of RAM, and Microsoft Windows XP operating system. The GA parameter settings are based on a small pilot study: population size equal to 50, probability of crossover equal to 0.90, and probability of mutation equal to 0.0015. The termination criterion is based on the runtime. Two minutes are allocated to the small terminal problems; 10 minutes are allocated to the medium terminal problems, and 20 minutes are allocated to the large terminal problems.

7.3. Results analysis

Table 4 shows the quality ratios for the sets of test problems. UGBSA provides the best solution results overall. The average solution results are 16% better than the solutions from EGBSA and 24% less than RAND. The robustness of the heuristic is clear because UGBSA outperforms the other heuristics for each terminal size. The table shows the performance of the heuristics to the lower bound. The overall average solution quality of UGBSA is only 17% from the lower bound, whereas EGBSA and RAND average solution qualities are 36% and 45% from the lower bound. For the small terminal problems, UGBSA produces average solutions that are only 5% to 8% from the lower bound. As the terminal size increases, the gap between average solutions of UGBSA and the lower bound increases. However, UGBSA still offers solutions that are better than those offered by the other methods.

Table 4 The experiment results for $\beta(q)$ and $\eta(q)$

Facility	Trailers	$\beta(q)$			$\eta(q)$		
		UGBSA	EGBSA	RAND	UGBSA	EGBSA	RAND
Small	18	1.00	1.11	1.27	1.05	1.17	1.34
	30	1.01	1.10	1.21	1.08	1.18	1.29
Medium	60	1.00	1.23	1.22	1.13	1.39	1.38
	100	1.00	1.18	1.23	1.12	1.33	1.38
Large	300	1.00	1.22	1.27	1.31	1.60	1.67
	500	1.00	1.14	1.23	1.31	1.49	1.61
Overall		1.00	1.16	1.24	1.17	1.36	1.45

Table 5 shows that UGBSA finds the best solution 94.3% of the time (283 times out of 300). There are only eight occasions where EGBSA finds the best solutions, which occur on the small terminal problems. RAND finds the best solution one time out of 300, which occur on the large terminal problems. Regarding the average computational times, UGBSA requires about the same amount of computational time as EGBSA requires, but with better performance. UGBSA requires about 1.6 minutes on average to produce solutions for the small terminal problems and about 12 minutes on average to produce solutions for the large terminal problems. Compared to the simulation-based scheduling approach in McWilliams et al. (2008) which requires nearly 360 minutes to find solutions that are less superior to the solution of UGBSA, it is easy to conclude that both EGBSA and UGBSA are better solution approaches to solve the PHSP. However, UGBSA is for the more general case of the PHSP (scheduling problem with unequal batch size inbound trailers).

Table 5 The experiment results for $\psi(q)$ and $CPU(q)$

Facility	Trailers	$\psi(q)$			$CPU(q)$	
		UGBSA	EGBSA	RAND	UGBSA	EGBSA
Small	18	48	4	0	1.61	1.44
	30	36	4	0	1.63	1.09
Medium	60	50	0	0	8.91	4.90
	100	50	0	0	6.61	4.93
Large	300	49	0	1	13.19	12.72
	500	50	0	0	10.72	12.61
Overall		283	8	1		

8. CONCLUSION

In this paper, the PHSP, a scheduling problem that is common in the parcel delivery industry, was considered. The problem consists of processing a large number of inbound trailers at a much smaller number of unload docks with the objective of minimizing the timespan of the transfer operation. The PHSP was modeled as a minimax problem. Because a minimax problem is known to be NP-hard, a genetic algorithm was developed to search for good solutions to the PHSP. The performance of the proposed algorithm was compared to the performance of an existing algorithm, random scheduling, and a lower bound. The results showed that the proposed solution approach offers solutions that are within 17% of the lower bound, 16% better than the existing algorithm, and 24% better than random scheduling. The study shows that scheduling can have a significant impact on the operating performance of a parcel consolidation terminal and that the cost tradeoff to generate unload schedules can be minimum compared to potential savings in labor cost.

As with McWilliams (2009), the major weakness of the proposed solution algorithm is that it is formulated as a minimax problem. The main drawback of the minimax formulation is the large number of non-unique solutions, resulting in a large solution space. Searching for good solutions in a huge solution space can be problematic for any search methodology. An improved solution approach should further reduce the computational time required for large size problems. The limitation of the solution approach is greatly depended on the parcel consolidation terminal configuration. However, many of the terminals tend to be configured similar to the one presented in Figure 1.

The future research for the PHSP should be to continually relax the stated simplifying assumptions in McWilliams et al. (2008). The future studies should consider related unload docks and load docks, the consideration of trailer replacement time, and the value of information on parcel sequence. The future research should also consider the application of tabu search and simulated annealing to solve the PHSP.

REFERENCES

[1] Bartholdi J., Gue K. (2000), Reducing Labor Costs in an LTL Crossdocking Terminal; *Operations Research* 48(6); 823-832.

[2] Basset M., Pekny J., Reklaitis G. (1995), Decomposition Techniques for the Solution of Large-Scale Scheduling Problems; *American Institute of Chemical Engineering Journal* 42(12); 3373.

[3] Felix Ammah-Tagoe (2006), Freight in America: A New National Picture; Research and Innovative Technology Administration, Bureau of Transportation Statistics; Washington, DC.

[4] Gilbert D. (1934), Parcel Conveyors; *Post Office Electrical Engineers' Journal* 27(3); 179-186.

[5] Goodison H. (1981), Parcel Sorting Systems and Ancillary Postal Equipment; *Post Office Electrical Engineers' Journal* 74(3); 271-277.

[6] Goldberg D. (1989), Genetic algorithms in search, optimization, and machine learning; Addison-Wesley: Reading; MA.

[7] Gould L. (1991), Increase Productivity by Choosing the Right Sortation Conveyor; *Modern Materials Handling* 46(7); 54-56.

[8] Graham R.L. (1969), Bounds on Multiprocessing Timing Anomalies; *SIAM Journal of Applied Mathematics* 17(2); 416–429.

[9] Gue K. (1996), *Freight Terminal Layout and Operations*; Doctoral dissertation, Georgia Institute of Technology, 1995; (Dissertation Abstract International, B56/11, 6322-6432).

[10] Haupt R., Haupt S. (1998), Practical Genetic Algorithms; New York: John Wiley & Sons.

[11] Khmelnitsky E., Kogan K., Maimon O. (2000), A Time-Decomposition Method for Sequence-Dependent Setup Scheduling under Pressing Demand Conditions; *IEEE Transactions on Analytical Control* 45(4); 638-652.

[12] Lai J. (2007), *Surrogate search: A Simulation Optimization Methodology for Large-Scale Systems*; Doctoral dissertation, University of Pittsburgh, 2006; (Dissertation Abstracts International, B67/08, 205-364).

[13] Masel D., Goldsmith D. (1997), Using a Simulation Model to Evaluate the Configuration of a Sortation Facility; *Proceedings of the 29th Winter Simulation Conference*; Atlanta, GA: IEEE Computer Society, 1210-1213.

[14] Masel D. (1998), Adapting the Longest Processing Time Heuristic (LPT) for Output Station Assignment in a Sortation Facility; *Proceedings of the 7th Industrial Engineering Research Conference*, Phoenix, Arizona: IIE.

[15] McWilliams D. (2009), Genetic-Based Scheduling to Solve the Parcel Hub Scheduling Problem; *Computers and Industrial Engineering* 56(4); 1607-1616.

[16] McWilliams D., Stanfield M., Geiger C. (2008), Minimizing the Completion Time of the Transfer Operations in a Central Parcel Consolidation Terminal with Unequal-Batch-Size Inbound Trailers; *Computers and Industrial Engineering* 54(4); 709-720.

[17] McWilliams D., Stanfield M., Geiger C. (2005), The Parcel Hub Scheduling Problem: A Simulation-Based Solution Approach; *Computers and Industrial Engineering* 49(3); 393-412.

[18] Rohrer M. (1995), Simulation and Cross Docking; *Proceedings of the 1995 Winter Simulation Conference*; IEEE Computer Society, 846-849.