Real-Time Dispatch of Petroleum Tank Trucks

Gerald G. Brown; Glenn W. Graves

Management Science, Vol. 27, No. 1 (Jan., 1981), 19-32.

MANAGEMENT SCIENCE
Vol. 27, No. 1, January 1981
Printed in U.S.A.

REAL-TIME DISPATCH OF PETROLEUM TANK TRUCKS*

GERALD G. BROWN† AND GLENN W. GRAVES‡

A highly automated, real-time dispatch system is described which uses embedded optimization routines to replace extensive manual operations and to reduce substantially operating costs for a nation-wide fleet of petroleum tank trucks. The system is currently used in daily operations by the Order Entry and Dispatch segment of the Chevron U.S.A. Marketing System. Refined petroleum products valued at several billion dollars per year are dispatched from more than 80 bulk terminals on a fleet exceeding 300 vehicles in approximately 2600 loads per day. Centralized use of the dispatch system required its design and implementation as a set of transaction modules within a large management information system. This environment presents special challenges for the optimization methods; an heuristic sequential network assignment was developed for certified performance on these dispatch models in lieu of their solution as integer programs. Objectives include minimizing transportation costs (approaching $100 million annually) while maintaining equitable man and equipment workload distribution, safety standards, and customer service, and satisfying equipment compatibility restrictions.

(PETROLEUM INDUSTRY; TRANSPORTATION, ROUTE SELECTION; INTEGER PROGRAMMING, HEURISTIC; INTEGER PROGRAMMING, APPLICATIONS; VEHICLE DISPATCHING)

1. Introduction

The methods described here have been developed to assist in the timely control, and economic use of a nationwide bulk delivery fleet of petroleum tank trucks. This portion of the system is intended to aid dispatchers by correlating large amounts of data in real time and producing nearly complete shift dispatches for each bulk terminal from which loads are hauled. The dispatchers, located at a central national order processing facility, must each handle several bulk terminals as well as coordinate other activities related to product availability, order entry, nonproprietary equipment requirements, and (recently) allocation.

Fundamental to the philosophy of the system is that the human dispatcher cannot be replaced. Rather, he must be materially assisted in his work by quick and comprehensive presentation of dispatch information in concise terms, with identification of exceptional conditions requiring manual intervention. The dispatcher is expected to make whatever adjustments are necessary while preserving the overall quality of the dispatch.

Very strict computer performance criteria must be met by the system. Even the largest dispatch should not require more than a fraction of a second of computer time or more than a very small memory region. This efficiency (as well as reliability) is vital to the effective use of the system as an integrated transaction module in a real-time information management system on a congested host computer. Daily operations require hundreds of trial dispatches during a very short time period, although this is mitigated somewhat by time zone differences across the continent.

---

*Accepted by Arthur M. Geoffrion, former Departmental Editor; received January 3, 1980. This paper has been with the authors 2 months for 1 revision.

†Naval Postgraduate School, Monterey.
‡University of California, Los Angeles.

0025-1909/81/2701/0019$01.25

The system is intended to reduce operating costs in several ways [3]. It controls overtime (and undertime) for vehicles and drivers, helps reduce costly human errors, and uses the most economic available means of transportation. Indirect objectives include greater manpower utilization system-wide, equitable distribution of workload, and other benefits.

The overall system can be viewed as processing, maintaining and displaying for each bulk terminal a large volume of information concerning the bulk terminal area, customer orders, and vehicles. These data are used to produce in a timely manner two shift dispatches per day for each terminal. Each dispatch must satisfy many explicit specifications of customer delivery times and mileages, special equipment requirements, product-specific capacities of each vehicle compartment, delivery time restrictions, and so forth. The dispatcher must also consider myriad implicit conditions such as rush hour traffic congestion, local road and weather conditions, adjustment limits on ordered quantities to suit available vehicle compartments, etc.

In the sections that follow, we introduce basic elements of the problem in sufficient detail to motivate key design decisions, propose an overall solution scheme for the dispatch process, formulate an integer linear program for the principal dispatch module, and describe implementation and system use. A concluding discussion considers what should, and what should not, be automated in the dispatch system.

2. Elements of the Problem

In this section, the basic elements of the dispatch problem are introduced in the context of daily operations. There is necessarily much simplification. However, the complex environment within which the dispatch function is embedded is both technically and organizationally relevant to the solution methods introduced later.

Bulk Terminals

Each bulk terminal acts as a storage point for as many as 16 products ranging from weed oil to jet fuel, but dominated in terms of sheer volume by motor gasoline. Product is received by pipeline, barge or truck, stored in tanks, and transferred to delivery vehicles via drive-through loading racks. Drivers are domiciled with company-owned vehicles at the terminal, with collocated service facilities for vehicle maintenance. Figure 1 shows the location of terminals for this system.

Delivery Vehicles

Delivery vehicles possess a wide variety of features relevant to their use in the dispatch. A model truck and trailer rig (see Figure 2) is equipped with multiple, isolated compartments. Each compartment has a volumetric capacity specific to the density of the product contained. Loading is accomplished at the bulk terminal from top or bottom outlets at a loading rack. Customer delivery is generally made by gravity feed with a valve manifold connecting the compartment via a hose to an underground storage tank; the entire content of each compartment is dropped, necessitating careful prior determination of available storage tank capacity to preclude accidental spills.

The variations of this basic vehicle design are myriad. They result from local vehicle laws, geographical and temporal demand patterns, and historical management policy. Each vehicle may have from 1 to 6 compartments, special fittings, meters and pumps, manifolding which prevents cross-product contamination (e.g., lead) upon delivery, vapor recovery gear, and so forth. Every truck must be loaded in accordance with its

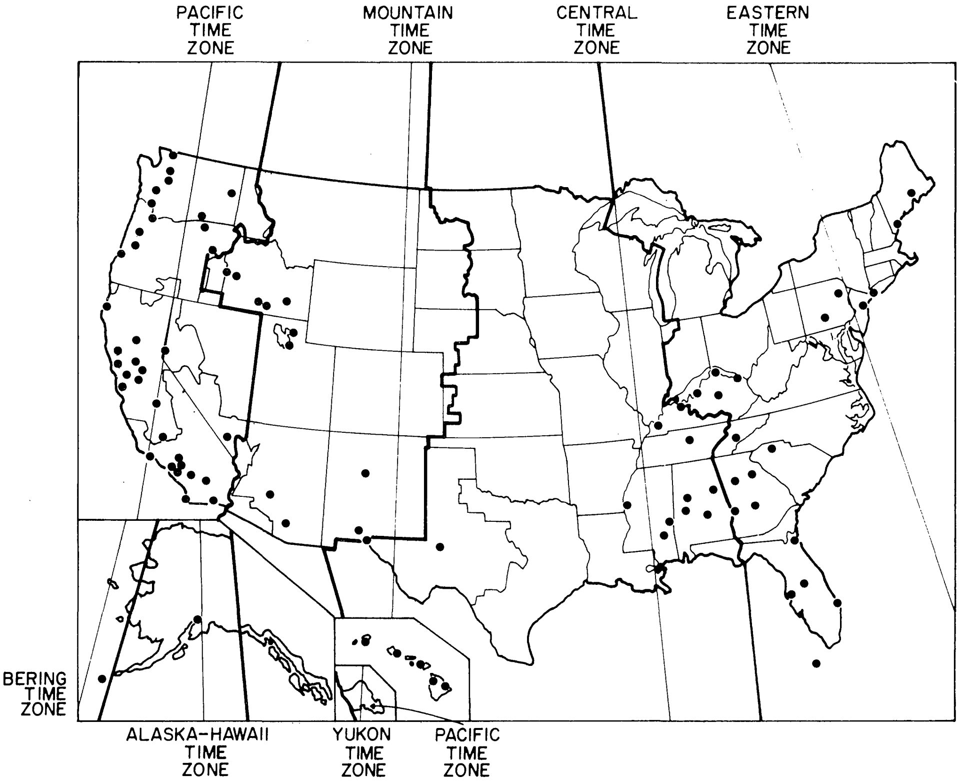
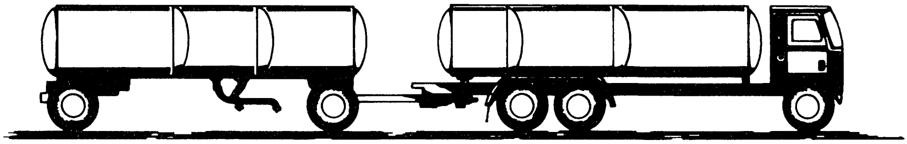


FIGURE 1. Bulk Terminals.

weight limits, those of road jurisdictions to be traversed, and such that compartments empty, or not fully loaded, satisfy a complicated specification arising from safe road handling considerations.

Vehicle operating costs are specified for each proprietary truck on a customer-by-customer basis as a function of mileage and standard delivery time. Nonproprietary truck costs may also be simple functions of actual delivery time and mileage, or may be fixed point-to-point charges for each trip depending upon operating region and contract terms and duration.

Each vehicle is assigned a sequence of loads for a shift with the duration of each shift determined by driver availability, vehicle availability, and contract terms. Overextension of vehicle shifts leads to overtime labor costs, disrupts following shifts, and can foment employee dissatisfaction. Underutilization of vehicles causes other similarly unfortunate outcomes, the most obvious being economic.



Typical Compartment Configuration	Gasoline: 2000, 900, 1700; 900, 1200, 2000
(Trailer aft; truck forward)	Diesel: 1500, 750, 1500; 700, 1000, 1700 (Gal.)

FIGURE 2. Delivery Vehicle.

Customer Orders

Each customer order is received by telephone at the nationwide dispatch facility, where an order processor immediately retrieves on a display console the customer's order file. Each order typically includes three products, usually grades of gasoline, jointly constituting a complete truck load. Following credit and allocation releases and other preliminaries, the order is entered into the information management system with the desired quantities of each product, the target delivery date and shift(s), and additional data regarding special equipment requirements (such as special couplings, pumps, an unmarked truck, and so forth), driver instructions, and billing data. The entire order entry process requires at most a couple of minutes, averaging 30 seconds.

Orders are accumulated throughout the day for future delivery. Soon after a cutoff time for acceptance of orders to be delivered on the following day, a dispatcher extracts on his display console all the orders to be satisfied, assesses equipment and driver availability at each bulk terminal, and determines bulk terminal area conditions such as available product inventory, weather, etc. Some initial adjustments may be made, such as: arranging for additional, non-proprietary vehicles, deferring excess orders to other bulk terminals, or to later shifts by delivery priority, and specifying additional delivery restrictions for some orders owing to safety or other considerations.

Finally, the complete dispatch is assembled and transmitted to the bulk terminal. Minor variations may be handled subsequently by the terminal, but any necessary major revisions are referred to the central dispatch facility.

3. Solution Scheme and Supporting Data

Our analysis of the dispatch function reveals that much of the time-consuming, detailed work naturally lends itself to further automation. However, not all details can be reasonably or economically integrated in an efficient manner.

The following is a general sequence of events for these dispatches:

1. Preview of dispatch. Extract customer order and vehicle data, review for special cases, balance general workload, insert new or missing information, etc.;

2. Compatible vehicle edit. Determine which vehicles can be used to deliver each order, considering equipment restrictions, compartmentation adequacy, etc.; but *not* transportation cost;

3. Assign orders to vehicles. A good dispatch minimizes operating costs while honoring vehicle and driver shift length restrictions;

4. Adjust order quantities. Order quantities may require adjustment to fit available vehicle compartmentation in an acceptable *filling sequence* (i.e., actual permutation of products in compartments) even for vehicles well suited to carry the load;

5. Final review. Identify any remaining exceptional conditions, and either perform minor adjustments manually or return to step 1 with modified conditions;

6. Issue dispatch. Print load documents for each vehicle shift at the bulk terminal site.

A critical review of this scheme was performed to identify opportunities to reduce manual workload or improve dispatch quality. Steps 1 and 5 heavily involve human judgment and do not invite much further automation. Steps 2 and 4, on the other hand, are time consuming and detailed, yet appear to be successfully manually performed with fairly simple rules of thumb. Of course, Step 3 offers the most obvious new opportunity for outright modelling, pursued in the next section.

The implicit decomposition of cost minimization and load sizing issues in Steps 2–4 has not been altered. Considering market policies and competitive environment, the coordination of these features is performed by indirect means over long run operation of the system; customers are encouraged to order product quantities which yield economic loads, and vehicles are designed and located to meet temporal variations in demand. In this way, maximum net profit of fitted loads gives way to equitable customer service at minimal transportation cost at the individual dispatch level.

Supporting data for each customer order in this idealized dispatch scheme include the products and quantities ordered, conditions on the degree of admissible adjustment in the quantities actually delivered, and delivery shift restrictions. In addition, each customer exhibits static data such as location, standard delivery time and mileage, and special delivery equipment requirements.

Each vehicle is statically described in terms of operating cost, compartment configuration and capacities, and special delivery equipment. For each shift, availability is specified; vehicles may be only available for a partial shift due to scheduled maintenance, Department of Transportation (D.O.T.) driver limitation, or other factors.

Finally, data specific to the bulk terminal includes fixed and variable transportation delay factors for deliveries, useful for reflecting effects of weather, loading equipment failures, and so forth, as well as product-specific densities, corrected for temperature, and penalities used to indicate which products are in short supply and thus invite equitable downward adjustment of delivered quantities.

4. An Integer Programming Formulation

This section develops and discusses an integer linear programming model which incorporates several of the coordination issues in a good dispatch, and is intended for use in Step 3 of the solution scheme.

Notation

Indexes

i	indexes orders;
j	indexes trucks;
$J(i)$	index set of trucks compatible with order i.

Given Data

c_{ij}	round-trip transportation cost of delivering order i with truck j;
t_{ij}	round-trip transportation time of delivering order i with truck j;
$\bar{s}_j, \underline{s}_j$	maximum and minimum shift lengths for truck j;
$\bar{z}_j, \underline{z}_j$	penalty cost rates for violating shift length restrictions.

Decision Variables

y_{ij} binary variable indicating whether or not order i is to be dispatched as a load on truck j;

x_j trivalent variable indicating the applicable elastic penalty $(\bar{z}_j, 0, \underline{z}_j)$ for shift lengths of any solution.

Integer Linear Program

$$\min_{y} \sum_{i} \sum_{j \in J(i)} c_{ij} y_{ij} + \sum_{j} x_j \left[s_j - \sum_{i} t_{ij} y_{ij} \right] \tag{1}$$

subject to

$$\sum_{j \in J(i)} y_{ij} = 1, \quad \text{all } i; \tag{2}$$

$$\begin{aligned}
&\text{(a)} \quad s_j = \bar{s}_j \quad \text{and} \quad x_j = \bar{z}_j \quad \text{when } \sum_i t_{ij} y_{ij} > \bar{s}_j; \\
&\text{(b)} \quad s_j = \underline{s}_j \quad \text{and} \quad x_j = \underline{z}_j \quad \text{when } \sum_i t_{ij} y_{ij} < \underline{s}_j; \\
&\text{(c)} \quad s_j = \bar{s}_j \vee \underline{s}_j \quad \text{and} \quad x_j = 0 \quad \text{when } \underline{s}_j \leqslant \sum_i t_{ij} y_{ij} \leqslant \bar{s}_j;
\end{aligned} \tag{3}$$

$$y_{ij} \in \{0, 1\}, \quad \text{all } i, j. \tag{4}$$

Note that the penalties satisfy $\bar{z}_j \leqslant 0 \leqslant \underline{z}_j$ by implication when $\underline{s}_j < \bar{s}_j$.

The objective function reflects the potentially conflicting desire to minimize operating costs while simultaneously honoring the shift length restrictions. The second term penalizes any undertime, or overtime for truck shifts.

Constraints (2) ensure delivery of each order as a single load.

Specifications (3) and (4) simply enforce the desired model composition.

The model was originally considered with rigid shift lengths (i.e., $\underline{s}_j = \bar{s}_j$ and $\underline{z}_j = -\bar{z}_j = +\infty$), expressing the widely professed belief that a good dispatch must utilize all equipment fully. Of course, no feasible solution can be guaranteed for such a formulation, as was reinforced by management review of many manual dispatches.

The model was first implemented with strict minimum and maximum shift lengths (i.e., $\underline{s}_j < \bar{s}_j$ and $\underline{z}_j = -\bar{z}_j = +\infty$). This permitted some flexibility in assembling feasible solutions, but it required inordinate preview of vehicles and orders in Step 1 to insure feasibility.

Finally, the *elastic* shift limits were provided as shown here. This formulation permits violation of shift length restrictions for each vehicle at a specified rate of cost. The penalty costs can be used to coerce prioritization of shift violations as an integral economic consideration. Considerable analysis has been invested in the specification of these shift limits, costs, and penalties for each vehicle, each bulk terminal, and each shift limit rationale (e.g. D.O.T. driver limits are much more inflexible than simple driver overtime).

Note that each order has been assumed to be a full truck load. Although customers are urged to order this way, there are still a few exceptions. These are either dispatched on small trucks, or consolidated with other small orders by the dispatcher for multiple-stop delivery. These *split loads* are not easily automated since no data is currently available for customer-to-customer travel times and mileages, and their frequency and value are too small to justify the initial investment in terms of reduced dispatcher workload.

5. Implementation and Use

An elastic integer linear programming procedure and supporting data were developed and improved over many months. Benchmarks for the prototype system were extracted from daily operations at several bulk terminals and used to compare offline system results side by side with actual manual dispatch performance. The early results were very encouraging.

Compared with manual dispatches, the system produces extremely uniform distribution of workload among vehicles with significantly lower transportation costs. For those cases in which shift limits are violated, the system gives the explicit economic rationale for this outcome, and consequently proves to have excellent face validity for dispatchers and management. In this respect, the system wins the competition without qualification.

The benchmarks have also produced some unexpected results. Some popular rules of thumb used by manual dispatchers in Step 3 prove to be very uneconomical. Further, the system relentlessly reveals undetected data errors (a distinct advantage of optimization) that have instigated major internal review of transportation cost and time standards. Finally, it has become clear that some bulk terminal areas are significantly more difficult to dispatch well than others. Surprisingly, it is much easier for the automated system to produce a good dispatch for a large terminal than for a small one.

Unfortunately, the conditions under which the system must operate are rather severe. Since the dispatch system must cohabit in real time with a large information management system in constant use, very little pure computational power remains. Worse, the architecture of the real-time computer system is totally oriented to a transaction-driven software package. Each transaction is expected to consume minimal resources—at most, a fraction of a second of computer time and a very small region. Overall performance considerations for the system do not permit large, heavily computational tasks to be performed without unconscionable delay in response time (either that for the originating dispatcher, or for the hundreds of other users of the system at the time). Even more, the operating system resource monitor expects transactions to consume increments of system resources in uniform, small, and predictable quanta.

This is not an ideal environment for integer programming.

The following representative benchmark illustrates the situation. This dispatch has 28 trucks and 103 orders, producing a model with 811 binary variables. Some orders can be carried by only one, or two special trucks, others will suit as many as 23 trucks; on the average, $J(i)$ has nearly eight entries per order. Standard delivery times vary across orders such that a typical vehicle shift may carry as few as one, or as many as ten loads. Among trucks, the standard delivery times and costs for any given order may differ by 50 and 250 percent, respectively.

Run	Conditions	Solution Quality	Solution Seconds
1	MPS	unknown	300 +
2	XS (Default)	2.1%	14.1
3	XS (GUB)	1.8%	6.4
4	XS (Tuned)	0.6%	3.0

Solution quality is the percentage by which the value of the integer incumbent exceeds the best lower bound at termination. *Solution seconds* are for IBM 3033.

Run 1 was performed with a commercial optimization system, which had difficulty possibly related to poor enumeration tuning (not pursued). All subsequent runs were with our X-System, an optimization system serving here as an experimental testbed [2].

Run 2 indicates initial performance with default tuning. Run 3 gives the results achieved by exploiting the Generalized Upper Bound [4] structure of the shift-length constraints. Run 4 shows the best performance achieved by problem-independent tuning of the x-System, which requires about 100K bytes region. Runs 2–4 were automatically terminated when solution quality met a specified tolerance of, respectively, 3, 2 and 1 percent.

This performance is *not good enough* for production use, especially with a workload of several hundred daily runs within a few hours during peak-load computer conditions, relieved only slightly by the dispersion of activities across six time zones (Figure 1).

A customized optimization system was mandated. Options considered, and rejected, included tailoring the general X-System for this particular model. The problem can be restated as a binary network assignment problem with gains, but the need to preserve elasticity features mitigates the usefulness of this observation. An alternate approach would be development of a network factorization algorithm [6]. Both avenues were investigated, with the conclusion that very little marginal improvement in performance would result. Analysis of algorithm performance predicts that there is not a significant difference between the work performed by the general X-System with standard basis factorization and by a network variant with full elastic and integer enumeration capability.

6. Efficient Heuristic with Embedded Optimization

At this point, and certainly with no misplaced sense of nobility, an heuristic was considered. First, sheer speed is of the essence. Second, the problems occur with regular structure in day-to-day operations at each bulk terminal, providing both an opportunity to develop site-specific tuning and a fairly reliable method to detect misbehavior. Finally, the model can be fully optimized for purposes of calibration and selective audit by the off-line optimization system already available.

The design of the heuristic draws from computational experience with the quadratic assignment model [7] and hybrids with linear programming [5], [8]. Also, the underlying network structure (exclusive of the gains and attendant floating point arithmetic and basis structure) invites application of a *pure network algorithm* embedded in the heuristic.

With this in mind, the following simple solution procedure was developed. A *sequence* of embedded network problems is generated and solved with a variant of GNET [1]. Each such solution is used to fix *some* of the orders as loads on trucks. This process is terminated when all orders are assigned, or when no further progress can be made.

The generic network problem is shown in Figure 3 and described mathematically below.

Notation

Indexes

i indexes unassigned orders, only (cardinality m);
j indexes trucks;
$J(i)$ index set of compatible trucks.

Given Data

k denotes a dummy order (e.g., $k = 0$)

l_j total of t_{ij} for orders already assigned as loads on truck j;

τ_j remaining truck time projection: $(\underline{z}_j \bar{s}_j - \bar{z}_j \underline{s}_j)/(\underline{z}_j - \bar{z}_j) - l_j$;

$t_{\text{inf}}, t_{\text{sup}}$ smallest, largest standard transportation times for unassigned orders;

c'_{ij} projected, penalized transportation cost;

$$c'_{ij} = \begin{cases} c_{ij} + \bar{z}_j(\tau_j - t_{ij}) & \text{if} \quad \tau_j - t_{ij} \leqslant 0, \\ c_{ij} - \bar{z}_j(\tau_j - t_{ij}) & \text{if} \quad 0 < \tau_j - t_{ij} \leqslant t_{\text{sup}}/2, \\ c_{ij} - \bar{z}_j(\tau_j - t_{ij} - t_{\text{inf}}) & \text{if} \quad t_{\text{sup}}/2 < \tau_j - t_{ij} \leqslant t_{\text{inf}}, \\ c_{ij} & \text{if} \quad t_{\text{inf}} < \tau_j - t_{ij}; \end{cases}$$

b_j maximum number of orders still assignable to truck j:

$$b_j = \begin{cases} 1 + \lfloor \tau_j / t_{\text{inf}} & \text{if} \quad \tau_j > 0, \\ 0 & \text{otherwise,} \end{cases}$$

($\lfloor$ indicates the next lower integer.);

u_j range of the number of orders still assignable to truck j:

$$u_j = \begin{cases} b_j - \lfloor \tau_j / t_{\text{sup}} & \text{if} \quad \tau_j > 0, \\ 0 & \text{otherwise;} \end{cases}$$

d total excess (unassignable) orders: $\sum_j b_j - m$.

Decision Variables

y_{ij} variable indicating whether or not order i is preferred as a load on truck j;

s_j variable indicating estimated surplus loads preferred for truck j.

Embedded Network

$$\min_{y} \sum_{i} \sum_{j \in J(i)} c'_{ij} y_{ij} \tag{1}$$

subject to

(sources)

$$\sum_{j \in J(i)} y_{ij} \leqslant 1, \quad \text{all } i; \tag{2}$$

$$\sum_j s_j \leqslant d;$$

(sinks)

$$-s_j - \sum_i y_{ij} = -b_j, \quad \text{all } j; \tag{3}$$

$$\begin{aligned} & y_{ij} \in \{0, 1\}, && \text{all } i, j; \\ & s_j \in \{\text{integer}\}, && \text{all } j. \end{aligned} \tag{4}$$

The units of this pure network formulation are increments of time approximately equal to the standard delivery time of the shortest remaining unassigned order. The formulation assumes that all unassigned orders require this time increment for delivery and that remaining truck time is always some integral multiple of this time increment. The objective function reflects the projected consequence of assigning any remaining order to a truck. Many options are available in forming this objective. Shown here is one approach with four cases for c'_{ij} respectively representing: (1) certain overload, (2) small underload for which no other order is likely to fit, (3) underload for which at least one other order might fit as well and (4) extreme underload. This objective is chosen in concert with the one-pass, nonbacktracking fixing scheme shown below.

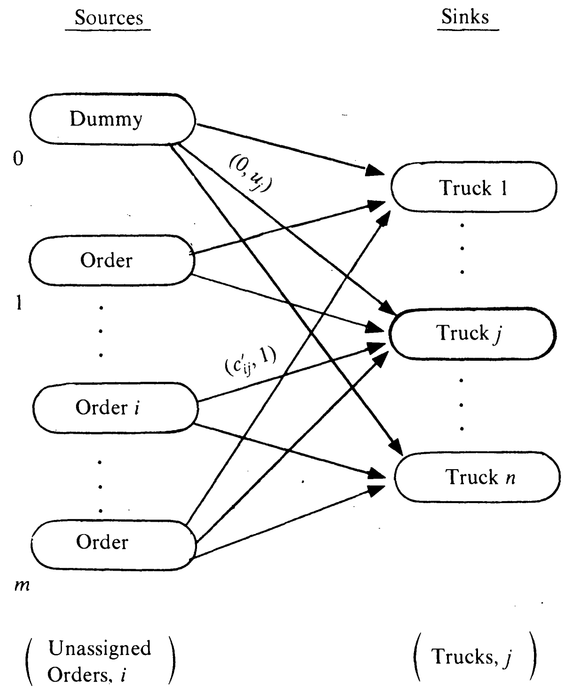


FIGURE 3. Embedded Network.

Each network solution is examined and orders are fixed as loads on trucks as follows. For each truck, if any orders have been assigned by the network solution, and if the assigned order with the longest delivery time does not create a worse total time penalty for that truck, fix that order and update the projected remaining truck time τ_j. Continue fixing orders for that truck until $\tau_j < T\,(t_{\text{inf}} + t_{\text{sup}})$ (e.g., $T = 2$, a heuristic parameter). When this condition is met, repeat the process for the next truck.

If at least one order is fixed for some truck, continue the heuristic with another (smaller) network problem.

When no order is assigned, the embedded network iteration ceases and any remaining unassigned orders are placed on a *spill list*. This list can include other orders orphaned by the compatibility edit in Step 2, and is treated as a phantom truck with transportation costs equivalent to the preferred short-term nonproprietary truck type for the local bulk terminal.

Next, the overall solution is improved if possible by simple pairwise comparisons and *switches* of loads between compatible trucks, or *slides* of loads from one truck to another. Higher order exchanges are not used.

7. Vehicle Compatibility and Order Quantity Adjustment

Following the fixing of orders as loads on trucks, the order quantities are adjusted for each load in Step 4 to produce the filling sequence that best suits the truck compartmentation. This typically involves assigning 3 products to 6 compartments. Each compartment is filled to its precise product-specific capacity, which is a function of temperature-corrected product density.

A direct enumeration scheme is used which determines which permutation of product-to-compartment assignments is most desirable. No adjustment is considered which completely eliminates a product or which violates certain adjacency restrictions. For each product, an "up" and "down" penalty per gallon adjustment is specified. For each load, order quantities of component products may also be coded with an additional class penalty per gallon representing varying degrees of inflexibility with respect to adjustment. Two optimal filling sequences are determined on the basis of total quantity adjustment penalty, one with adjacent compartments for each product, and one with no adjacency restrictions. The contiguous filling sequence is selected unless the companion sequence is significantly better by a constant specified by the dispatcher.

This scheme gives the dispatcher the capability to influence several overall factors for each bulk terminal. Quantity adjustments can be controlled in keeping with product supplies (especially shortages) so that customers are still equitably served. Contiguous product sequences are desirable because of the reduced driver workload at the loading rack and the customer site.

The success of the order quantity adjustment in Step 4 is highly dependent upon the compatible vehicle edit in Step 2, which restricts candidate vehicles for each order. On the other hand, to the degree that Step 2 is increasingly selective in screening compatible vehicles, the potential transportation cost savings of Step 3 are reduced. Balancing these effects, Step 2 is a compromise which is suggested by examining manual dispatch procedures.

The compatible vehicle edit of Step 2 examines each order with respect to all candidate vehicles indicated feasible for delivery. Initially, candidates are ruled out for obvious reasons (e.g., fewer compartments than products ordered).

At this point, Step 2 could be patterned after the quantity adjustment in Step 4, evaluating for each candidate vehicle the most desirable filling sequence to fit the order as a load, and editing vehicles with respect to a maximum permissible adjustment threshold. This approach is prohibitively expensive when applied to *all* candidate vehicles for *each* load.

Instead, a simple heuristic is used. Each candidate vehicle is ruled out if its *estimated* capacity is not sufficiently close to the total order quantity. (Recall that *actual* capacity is a function of product assignments to compartments.) Capacity is estimated by assuming that the entire order consists of the product with the largest order quantity, and "up" and "down" adjustment limits supplied by the dispatcher for each bulk terminal are applied. Next, if the *smallest* product order quantity is below a given volume, it is fit to the closest compartment on the candidate truck, and the truck is ruled out if the required adjustment is above specified limits.

8. System Performance

Steps 2, 3, and 4 can each be overridden by the dispatcher, who can specify compatible vehicles, fix loads, and assign compartments as he sees fit during the dispatch. This is especially useful for multiple iterations of these steps. For instance, delivery time restrictions occasionally require manual intervention for assignments made in Step 3.

The various parameters, penalties and limits of these procedures are specified for each bulk terminal. They are designed for easy comprehension and use by dispatchers in controlling the automated dispatch module, adapting to special local conditions, and responding to overall judgment on dispatch quality.

For the representative test problem cited in Section 5, the dispatch module requires 61K bytes and 0.2 compute second for the heuristic solution to the integer model in Step 3. Step 2 requires 0.1 second to edit compatible trucks and Step 4 requires 0.2 second to adjust order quantities.

The contracting network sequence in Step 3 requires a total of 5 steps, respectively with 811, 302, 154, 71 and 14 binary variables. The solution quality, compared to the earlier known bounds, is 0.5%. These results are very typical.

Solution quality can also be compared with bounds developed without optimization directly from problem data. For instance, if each order is assigned as a load on the cheapest (or most expensive) compatible truck, lower (or upper) bounds are derived for total transportation costs. With respect to this lower bound, the solution cited has quality 1.0%.

Many other performance measures are easily applied without resorting to outright optimization. For instance, an "ideal economic fleet" configuration is derived with and without compatibility restrictions by simple selection of cheapest transportation cost for each order, ignoring shift limits; this is used to evaluate selection, configuration and placement of trucks, to monitor the effects of encouraging customers to place standardized orders, and to reveal systematic errors in transportation cost and delivery time estimates.

Among the inevitable problems attending installation, the heuristic has required most tuning and modification to cope with extremely small dispatches—almost all design work centered on meeting performance criteria for big terminals. As dispatchers have come to increasingly depend upon the system, small nuisances have loomed with unforeseen significance. For instance, the greatly increased workload has made fleet sizing in Step 1 a bothersome task, which is now being studied for automation.

After many thousands of production runs and numerous minor adjustments, the dispatch module produces excellent solution quality and face validity with extremely reliable performance and high efficiency. In fact, most dispatches no longer require any adjustment or reruns at the final review, Step 5.

Improvements in operating efficiency have been impressive since adoption of the management support system which uses the dispatch modules. For instance, individual dispatchers now have the capacity to deal with up to 400 loads per day, compared with an industry average performance of 80–150 loads per day. In addition, transportation costs have been reduced by about three percent.

9. Conclusion

This project has provided many valuable lessons for both the managers and management scientists involved. The most fundamental decisions concern neither models nor implementation details. The crucial analysis focuses on what should and what should not be automated, and on how much compromise of reality is desirable in the automated portions of the system.

The environment for this particular work—a congested computer system, peak production workload, and capacitated personnel—has given an excellent aggregate means of evaluating results. The system would either provide better overall productivity, or fail completely. Fortunately, the quality, economy, efficiency, and face validity of the semi-automated dispatch solutions have been excellent, and the project is successful.

Individual productivity is increased only to the degree that the dispatcher still controls, understands, and accepts the automated modules. Most important, human judgement must be introduced naturally in such semi-automated systems to cope with extraordinary conditions. The total cost of automating responses to exceptional circumstances extends far beyond the solution modules in the host organization, and can render an otherwise desirable system totally infeasible. On the other hand, dispatchers have contributed some of the most insightful enhancements of the system *after* accepting and using it.

From the perspective of contemporary management support systems, there are continually increasing opportunities to apply optimization. However, classical optimization systems and techniques are rarely designed for use in the demanding, real-time environment so common to pervasive information management systems. Enforcing a monadic view of optimization, especially for combinatorial problems, reveals weaknesses not contemplated before by designers of stand-alone algorithms and systems.

For this particular problem, the environment and implementation schedule has mandated the use of heuristic methods. Heuristics are usually based on repeated applications of simple functions (such as sorting), just as they are often patterned after concepts useful in productive reasoning (such as greed). Fortunately, the remarkable efficiency of minimum cost network algorithms has recently made this class of model also available as such a routine tool. Methods employing nested sequences of conditional network problems show much promise for a wide range of combinatorial models, especially for those with embedded network structure such as the quadratic assignment model. Better yet, applications such as this encourage development of reliable, efficient techniques in a design discipline which may help make embedded optimization much more desirable for timely use on other important management issues.

As for solution quality, heuristics have a well-deserved reputation for unreliability. However, there are appropriate arenas for heuristic methods, especially if bounds can be developed for objective assessment of solutions. Bounded heuristics can serve admirably, reliably extracting much of the information that an algorithm would provide and producing solutions whose repetitive nature and audited error distribution can be shown to yield reasonably good results.

Adoption of this tactical dispatch system has presented new strategic opportunities. Among these, regional assignment of customer orders to bulk terminals, vehicle relocation, and even pipeline scheduling are now possible with a global perspective lended by the up-to-date, underlying high-resolution data bases now capitalized. Some of these issues are already under analysis by various support groups.[1]

---

[1] Special thanks are due to Gordon Bradley for his invaluable help, to Richard Lambeck and Marshall Mustain, Chevron U.S.A., Inc., for their continuing support of this project, and to Richard Haefele, Brian Putt and Gordon Topham, Standard Oil of California. We mourn that this is our last manuscript prepared by Rosemarie Stampfel.

References

1. Bradley, G., Brown, G. and Graves, G., "Design and Implementation of Large Scale Primal Transshipment Algorithms," *Management Sci.*, Vol. 27, No. 1 (1977), p. 1.
2. Brown, G. and Graves, G., "Design and Implementation of a Large Scale (Mixed Integer) Optimization System," paper presented at ORSA/TIMS, Las Vegas, Nov. 1975.
3. Chevron U.S.A., Inc., *Project Work Order* (authorized by General Manager of Marketing Operations, April 6, 1977).
4. Dantzig, G. and Van Slyke, R., "Generalized Upper Bounding Techniques," *J. Comput. System Sci.*, Vol. 1 (1967), p. 213.
5. Geoffrion, A. and Graves, G., "Scheduling Parallel Production Lines with Changeover Costs: Practical Application of a Quadratic Assignment/LP Approach," *Operations Res.*, Vol. 24, No. 4 (1976), p. 595.
6. Graves, G. and McBride, R. "The Factorization Approach to Large-Scale Linear Programming," *Math. Programming*, Vol. 10 (1976), p. 91.
7. ——— and Whinston, A., "An Algorithm for the Quadratic Assignment Problem," *Management Sci.*, Vol. 17, No. 7 (1970), p. 453.
8. Sibre, C., "A Ship Scheduling Model for the United States Coast Guard: a Quadratic Assignment/Linear Programming Approach," M.S. Thesis, Naval Postgraduate School, June 1977.