Chapter 5
Railway Blocking Process

Carl Van Dyke and Marc Meketon

5.1 Introduction and Background

Prior to the widespread adoption of unit trains and the rise of intermodal, most traffic moved in "loose car" or "manifest" service (also called "car load traffic"). In this type of service, sets of railcars are grouped together on a temporary basis into "blocks."

> A *block* is a group of cars that may have disparate origins and destinations, but will be moved together as a group from a common assembly point to a common disassembly point. At the disassembly point the block will be broken apart and the railcars will be formed into new blocks along with other railcars arriving from other locations. Thus, for an individual railcar, the origin and destination of a block may be either the same as the ultimate origin or destination of the railcar, or may be intermediate points in the railcar's route where the car is to be marshaled.

These blocks are moved by trains, where each train may carry a single block, or may carry multiple blocks. In this manner the railcars are relayed from their origin to their destination by being placed in a series of blocks, which are moved by a series of trains.

> In this context, a *marshaling or blocking plan* is the set of rules governing which blocks will be made at each location, and which cars will be put in each block.

Thus, the two main decisions in the design of a blocking plan are:

- The overall blocks to be created at each location.
- The specific traffic that should be placed into each block.

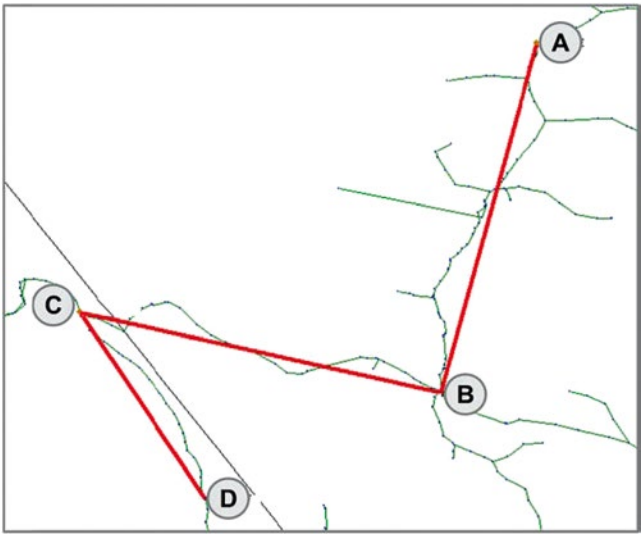
C. Van Dyke (✉)
TransNetOpt, Princeton, NJ, USA
e-mail: carl@cvdzone.com

M. Meketon
Oliver Wyman, Princeton, NJ, USA
e-mail: marc.meketon@oliverwyman.com

B.W. Patty (ed.), *Handbook of Operations Research Applications at Railroads*,
International Series in Operations Research & Management Science 222,
DOI 10.1007/978-1-4899-7571-3_5

5.1.1 Impact of Blocking on System Efficiency and Service

The efficiency of a railroad's production system for carload traffic is underpinned by the quality of the blocking plan. To understand this, let us first consider the routing of an individual shipment using the blocking plan. For this discussion, please examine the "block route" map shown below:



In the above figure, a shipment moves on a series (or "sequence") of three blocks: A–B, then B–C, and finally C–D. If we think about the process of moving the shipment, it would be something like the following:

Activity	Location	Time impact driver
Shipper release at A	A	N/A
Pick-up car from shipper	A	Train schedule
Process car into block to B	A	Yard processing
Wait for train to B to depart	A	Train frequency
Arrive at B	B	Train schedule
Process car into block to C	B	Yard processing
Wait for train to C to depart	B	Train frequency
Arrive at C	C	Train schedule
Process car into block to D	C	Yard processing
Wait for train to D to depart	C	Train frequency
Arrive at D	D	Train schedule
Process car for delivery	D	Yard processing
Wait for delivery time to customer	D	Train frequency
Deliver to customer	D	Train schedule

While the train schedules influence the transit times, and can have some impact on the routing of the blocks, the blocking plan determines where the shipments will be handled, and the aggregate or overall routing of the shipment. Various analyses of carload shipments show that shipments can often spend more time in yards being processed and waiting for trains to depart, than in actual transit on trains (Little et al. 1992). Thus, the blocking plan strongly influences the efficiency and service level that shipments experience by determining where and how often shipments will be handled, and how direct the overall routing will be.

The core influences of the blocking plan can be summarized as follows:

Service levels: each handling (classification) of a shipment represents a delay in the forward progress of the shipment. If you consider a typical North American example of having one departure per day for each block based on the train schedule, an 8 h processing time for cars, and a perfectly random arrival pattern for the cars being placed in each block, then the average time in the yard would be [Processing Time] + [Headway]/2, where headway is the time interval between departing trains carrying the block. Using our example this would be 8 + 24/2, or 20 h of delay every time a car is handled.

Reliability: each handling (classification) of a shipment represents an "opportunity" for a failure, where failure is defined as the shipment not departing on the expected outbound train at the expected time. There can be many causes of such failures, for example they could be due to late in-bound train arrival, a lack of timely processing of the shipment into the out-bound block, miss-classification of the shipment, a problem with the out-bound train, detection of a mechanical defect in the railcar, or a lack of capacity on the outbound train (Kwon et al. 1995). Based on the author's experience, it has been found that such failure rates for connecting to a specific outbound train can exceed 20 % at major yards. These failures introduce variability into the overall transit time and thus adversely impact the product quality experienced by the shipper.

Circuity: shipments do not always take the most direct path from their origin to their destination. The difference between the most direct path and the actual path represents the excess distance the shipment travels, which is called circuity. Circuity can be introduced by both the blocking plan and by the train plan. Arguably, the largest source of such circuity is the blocking plan. Because the processing of railcars into blocks benefits from economies of scale both in the overall processing of the cars and in the ability to form larger blocks going longer distances, the author's direct experience indicates that it is often the case that shipments are taken out of route to reach larger yards. In other cases, the initial or final movement of the shipment may require an out-of-route local move to reach the origin or destination "serving yard" for the shipment. For these and other reasons, circuity may be introduced, and this circuity is often determined by the design of the blocking plan.

Yard workloads: the blocking plan determines where shipments will be handled. This means that the blocking plan determines the workloads at yards, in terms of both the number of blocks being made, and the total number of railcars being

processed. The selection of which yards should perform which actions based on the blocking plan will thus determine the cost drivers for the yards, and can also influence the capital investments needed. The blocking plan modeling or design process can also be used as a tool for determining if specific yards can or should be closed or downgraded, and whether benefits would accrue from the upgrading of existing yards or the opening of new yards.

5.1.2 Specifying the Blocking Plan

Most blocking systems are location-based, and strive to be consistent—that is to provide the same instructions to all railcars or shipments with similar or identical attributes.

It is the author's understanding that before computers, blocking instructions were maintained in written form at each yard. Based on a railcar's destination, and perhaps a small number of other attributes or special conditions, a clerk could look up the block assignment in a paper blocking guide, and determine how to route the railcar.

Blocking plans were computerized well before the ability to create a computer-generated trip plan or car schedule was developed (see Chap. 4), with the Southern Pacific TOP system, and Missouri Pacific TCS systems being among the best examples (IBM; Railway 2014). This computerization process focused on converting the idea of the location-based blocking book into a similar set of location-based computer rules. These systems were enhanced by allowing a large set of shipment attributes to be considered when selecting a block for a shipment. This created a double-edged sword, simultaneously providing a great deal of control over how shipments were routed and greatly increasing the potential complexity of the rule sets.

The vast majority of railroads worldwide use some type of location-based, rules-driven, blocking look-up tables. The one major exception is the concept of using an algorithm for the generation of the railcar (shipment) to block assignments. Such an algorithm was developed by Norfolk Southern, and is now also used by Canadian Pacific Railway through adoption of the NS system (Norfolk Southern Corporation). It is the author's understanding that the development of an algorithmic capability is also under consideration at several other railroads as well. The algorithmic approach still uses rules, but also relies on business logic that takes a network perspective to determine the best or lowest cost sequence of blocks to use for each shipment. On the one hand, algorithms can increase the ease of plan maintenance, and allow for faster changes to the plan. On the other hand, algorithmic blocking can be more challenging to manage, and the user may have less control over the routing of specific shipments.

While concepts of "dynamic blocking" exist (Kraft 1998; Norfolk Southern Corporation), at most railroads the blocking plan is fairly static, and rarely changed on a "real-time" basis. This is true of both the algorithmic and table-based systems.

The concepts of table-based and algorithmic-based blocking are explored in more detail below, and the dynamic blocking concept is explored further in Chap. 4 on car scheduling. The authors have had direct experience with the design and data contained in the blocking systems at about a dozen major railroads in North America, Europe, Asia, and Africa (plus numerous smaller railways), as well as several planning systems and modeling tools for the design of blocking systems. The discussion that follows is based largely on this first-hand knowledge.

5.1.3 Plan Complexity

In the simplest approach to blocking, the final destination of each traffic record would be the sole determinant of which traffic should be placed in each block. Thus, the blocking plan specification would be based on a single attribute—final destination. However, for a variety of reasons, railroads use many other attributes to determine which shipments go in each block, greatly increasing the complexity of the blocking plans. A typical railroad will use between 20 and 30 attributes on a regular basis in assigning shipments to blocks.

Examples of reasons why these additional attributes are used, and specialized blocks are created include:

- Service differentiation—alternate blocks and trains are often provided for specific types of traffic such as intermodal, automotive or grain traffic, or to separate out unit train traffic.
- Restricted routings—some traffic must take specific routes due to safety considerations or to avoid damage to selected commodities. For example, some railcars may be speed-restricted, have clearance restrictions (cannot use some routes due to the height of bridges or tunnels), contain hazardous materials that must be taken over specific routes, or contain commodities that should not pass through a hump yard due to potential for damage.
- Interchange blocks—traffic bound to an interchange point with another railroad may need to be separated out by destination on the receiving railroad in order to improve service. In some cases there may be more than one receiving railroad at an interchange, and separate blocks may need to be made for each.
- Local blocking—traffic destined to the same station may need to be broken out by customer at the station, or by different parts of a customer's plant based on commodity.
- Empty blocking—in some cases empty railcars are routed differently, or the railroad wants to group the empty cars together (by car type) in order to expedite the movement of these empties to customers for loading.
- Specialized services—for a variety of reasons railroads enter into commercial agreements with customers to provide various specialized services that require shipments to be handled in a specific manner, and this results in the creation of specialized blocking instructions.

The end result of the above considerations, and a variety of others, is that the rules for specifying the blocking plan can become quite complex. While the majority of traffic may use fairly simple, destination-based rules, the level of effort for specifying and maintaining the plan becomes driven by these specialized blocking rules.

5.2 Current Industry Practices: The Blocking Rules Concept

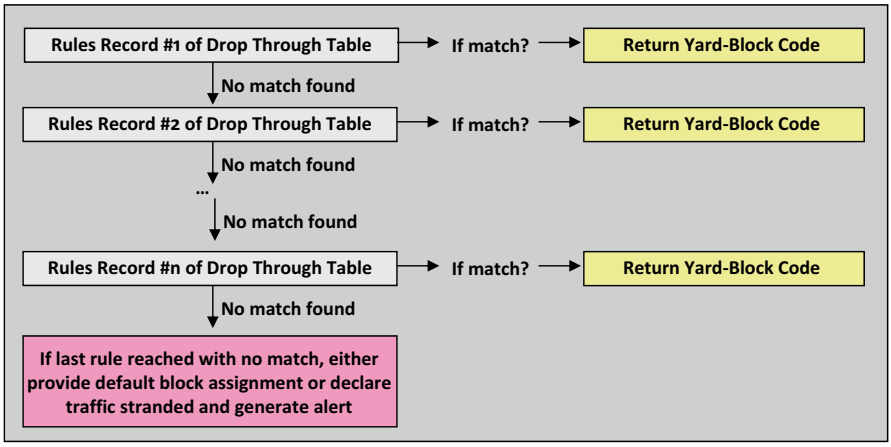
As noted above, most railway production blocking systems use a set of rules for determining which shipments should go in which blocks. These assignment processes are based on the current location of the railcar, and a variety of attributes related to the shipment and the railcar itself. Included in this process are a variety of special cases, which are discussed later in this section including block swaps, interchange blocks, and local blocking.

These rule or table-based blocking systems work as follows:

1. A set of rules are maintained for each location in the railroad network where blocking instructions must be generated. These rules are used for determining which block a particular shipment will be assigned to.
2. A request is made of the system to identify the block for a specific shipment. The blocking system is passed the current location of the shipment, the online destination for the shipment, and a variety of data on the overall shipment, the physical railcar being used, the content of the railcar, and the current status of the shipment.
3. The system processes the request by obtaining the blocking rules for the shipment's current location, and looking through those rules for the best match among the available blocks based on the information the system is given about the shipment.
4. The system returns a blocking code that it obtained based on its analysis of the rules for the current location.

While the details vary, all of the table-based systems work in a similar manner.

Central to the table-based blocking systems in widespread use is the concept of a drop-through rules table. In this type of table, the system starts with the first record in the drop-through table and tries to match the current shipment record to the criteria on the record in the drop-through table. If it matches, then the corresponding yard-block recorded on the record is returned. If it does not match, then each subsequent record in the drop-through table is checked until a match is found.



Each rule is typically composed of a series of attributes that the shipment must match in order to be assigned to the yard-block. As noted above, the rules are maintained by the planner or through business logic in a specific order, and the first rule that matches is used, thus ending the search. The rules are generally organized by location or a small group of locations, so only those rules that apply to the shipment's current location are considered.

Some railroads do not support the manual ordering of the rules, but instead use business logic to order the rules. This approach has been observed by the authors at two major international railways. In these cases, the rules are typically ordered by complexity, where the rules with more attributes specified come before the rules with fewer attributes. Priorities are then assigned to the attributes to further order records with the same number of attributes specified. In some cases, the user may also be able to specify rule priorities to change the relative order of the rules.

Based on the author's experience, most railroads use in the range of 20–30 separate attributes in their rules systems. Each rule typically uses only a small number of attributes. The assumption is that the values of all of the non-referenced attributes do not matter, and the shipment can have any value for those attributes during the matching process for that rule. By putting the rules in a specific order, lower rules can take advantage of the filtering effects of the prior rules. For example, consider the case where we want intermodal cars destined to location X to go in one block, and all other car types destined to X to go in a different block. We would specify the first rule as requiring all cars of car type P, Q, or S (intermodal car types) with a destination of X as going in block one. The second rule would simply say that all cars with a destination of X should go in block two, taking advantage of the fact that we already siphoned off the intermodal cars into a different block.

At a large yard, there can be several hundred blocking rules, and in some cases over a 1,000 rules. A smaller yard that only makes one or two blocks may have very few rules. The interactions and ordering of the rules are critical, and as a result the rules are generally maintained by a small number of highly trained individuals.

The rule attributes can generally be broken into several groups that include:

- *Primary traffic destination attribute*: There is generally a single destination code that is treated as the primary traffic destination attribute. The set of possible primary traffic record destinations that can be carried by a block is present in almost all rules. Several railroads organize their rules so that the rules at the end of the drop-through list are made up of only primary traffic destination attributes and each traffic destination appears exactly once among all of the destination-only rules emanating from a given yard. This ensures that all traffic will be routed. Because other attributes (as listed below) are also used in routing the traffic, the same primary destinations may appear multiple times across the more complex rules containing a mix of attributes. The primary destination codes are usually coded as stations within the railway, and are not coded macroscopically as city/state or microscopically as zone-track-spot. Destinations outside of the railway network typically have a predetermined interchange location that is used as the primary traffic destination code.
- *Shipment attributes*: these typically include the origin of the shipment, the destination, and the customer. Each of these pieces of information can be broken into a variety of separate pieces of information. For example, the offline origin can include the origin city/state, the origin SPLC code, the originating railroad, the railroad delivering the car to the railroad currently marshalling the shipment, the interchange received location, the online zone-track-spot data, etc. Similar details will exist for the destination, and the customer may be described in terms of both a code and a name, with distinctions made between the shipper, the consignee, the entity paying the freight bill, and the legal owner of the freight.
- *Railcar attributes*: the most commonly used attributes include the car type, the car's plate size, height, length, tare weight, the car's initial, the car's owner, whether the car is system, foreign or private, and the pool the car may be assigned to. In some cases, blocking systems can also use the car number.
- *Content attributes*: the most commonly used attributes are the net weight or gross weight, the load/empty status, and the commodity in the car. The commodity is typically expressed in terms of the STCC code or an internal, railroad commodity designation. For some commodities such as hazardous materials a special version of the STCC code may be used, or specific routing instruction codes may be applied. In some cases there may also be codes related to customs clearance, oversized dimensions, or other special considerations. For empties, the previously loaded commodity is often identified. Even the load/empty status can come in multiple flavors on some railroads.
- *Current status attributes*: this information relates to specific information on the status of the car at the moment that the classification request is made. Typically this consists of some combination of the current location of the car, the train it arrived on, and the yard-block or train-block it was on when it arrived at the current location.
- *Other attributes*: a variety of other attributes are used at various railroads. Examples include special codes specifying the run-through block the car is to be placed in (when going to a different railroad), routing instructions such as a

requirement that the car be weighed or cleaned somewhere in its routing, or a requirement to make intermediate stop offs. Another example is cars that have a mechanical problem and must be sent for repairs.

A variety of special cases can be handled. Two common examples are the blocking for local services and cars that do not have a destination specified. In some cases, cars that do not have a destination must be placed in "hold blocks" for manual handling, and in others "flow rules" are used to advance the cars in what is considered to be generally the right direction.

From a modeling perspective, the above approach represents three significant challenges:

1. Most production systems use a rules-based approach. If an algorithm is used in the planning process, how does one translate the results into a table-based form that can be used to direct the flow of the shipments?
2. The wide range of attributes and decision criteria reflect the overall complexity of the problem and the need to view the blocking problem from a multi-commodity perspective. This ultimately represents a huge challenge in the design and use of optimization or algorithmic strategies to design and improve blocking plans.
3. Given the nature of a rules-based system, significant challenges can arise in trying to improve the plan or even make modest changes. For example, determining what traffic should use a new yard can be very difficult because under a table-based approach, traffic will not naturally "flow" to a new yard, no matter what its cost.

Solutions to these problems and others will be explored further in this chapter.

5.2.1 Yard-Blocks, Train-Blocks, Class Codes, and Block Swaps

To provide some additional context for railroad blocking systems, there are a number of additional concepts that need to be understood, including the ideas of a *class code*, *yard-block*, *train-block*, and *block swap*.

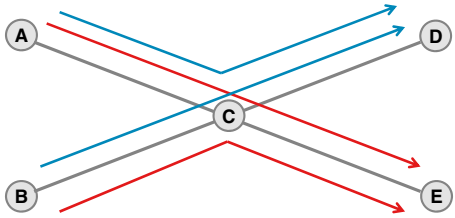
Perhaps for historic reasons, most blocking systems do not provide a definition of a block assignment in terms of a block origin, destination, and block name. Instead, they provide a "yard-block code," which is variously referred to as a "tag" or "class code," or in the case of CSX Transportation an "IYSC" or "inter-yard switching code." This "class code" is simply a name for the block, and does not specify where the block is going (except to the extent that the name matches a physical location on the railroad). The exception to this is the Norfolk Southern/Canadian Pacific system, which produces a full block definition including a destination. In most systems, trains specify a separate concept called a "train-block" that provides the pick-up location for the block, the set-off location, and a block name.

Class codes are then associated with the train-block. Since the class codes do not have a destination, the destination becomes the location where the train-block is set off. On the one hand, this makes it very hard to validate that appropriate class codes have been assigned to a particular train-block; on the other hand, it provides flexibility to send the same class code/yard-block to different locations by day-of-week or based on other factors related to the available train service.

In many cases, railroads assign multiple yard-block codes to a single train-block. This is often done to give visibility to subsets of cars within a block. There are at least three common reasons for doing this:

- One reason is to provide information on special cars in a block. For example, if a block contains some "hot traffic" that must be protected in the classification process at the next yard, this traffic can be identified by giving it a unique class code.
- A second reason is to provide visibility to the classification work that must be done on a block when it arrives at a yard. For example, for a local block arriving at a serving yard, one might show the out-bound class codes from the serving yard for each car on the in-bound block.
- A third is to allow easy "swinging" of traffic for capacity management purposes. For example, in a block going from yard A to yard B, one might segment the traffic into several groups. One group would be traffic that is absolutely best served by going to B, and there might be two other groups, traffic that would do all right going to yard C, and traffic that would be handled satisfactorily if it went to yard D. If yard B becomes congested, one could then easily divert some traffic to yard C or D by "swinging" the appropriate class code(s) to a different train-block that was going to C or D.

One consequence of the class code-based approach is that many railroad production systems do not know the intended destination of a class code or yard-block, making validation and support for block swaps difficult. A block swap is a situation where a train-block is passed from one train to another on an intact basis with no switching of the individual railcars within the train-block. Block swaps are done primarily for operational reasons in situations where the scheduling of a through train is not practical. The classic example of a block swap is shown in the figure below:



In this example, yard A makes a block for yards D and E, and yard B makes a block for yards D and E. In the simple case that each of these four blocks is only large enough to fill one-half of a train, a block swap provides a potentially desirable operational option. Yard A would create a train A–E for yard E, and yard B would create a train B–D for yard D. From their origins, each train would carry both a D and an E block, meaning each train would be full. At yard C, the A–E train would set off its D block, and pick-up an E block set off by the B–D train. Likewise, at yard C, the B–D train would set off its E block, and pick-up a D block set off by the A–E train. In this manner, a full switching of the railcars at C is avoided (minimizing work and potentially dwell time), and each train can operate to its capacity from origin to destination.

For the class code-based blocking systems, where the destination of the block is implied by the set off of the trains, block swaps represent a significant challenge. These systems often pass all shipments through the classification or blocking system every time a set off occurs. This could cause the shipments in the block swapped blocks to have their class codes or block assignments changed by the system (in our example above, by processing the cars on the D and E blocks through the classification system at location C). To counter this, many railroads end up putting in extensive instructions into their blocking systems to identify block swaps at the block swap locations, and ensure that the block assignments are protected as part of block swaps. Typically the systems use the in-bound train number, in-bound class code or yard-block, and other factors to create rules to ensure that the out-bound class code or yard-block remains the same.

These block swap rules are very difficult to maintain, and can be a significant nuisance during the creation of planning systems. In general, we will not address this issue further in this chapter. Most optimization systems and algorithmic blocking systems have full visibility to the block destination, which greatly simplifies the block swap issue—a block swap is assumed anytime a block is set off short of its destination.

5.2.2 Local Service

The usual notion of a block is that it is a group of cars that are assembled at one yard, which is then transported and delivered intact to a disassembly point. Specification of local blocks represents a challenge in that many distinct blocks or grouping rules may be required to identify all of the required, customer-specific groupings. One approach is to fully enumerate each customer block in full detail, resulting in numerous individual blocks with very specific rules. An alternate approach used by some railroads is to organize the blocking or routing information around the local services that will directly pick-up and deliver cars to customers. In this second approach, rather than specify individual blocks to each customer, a range of stations or customers may be specified instead, tied to the specific local service that will serve the stations or customers. In effect, the individual local blocks become implicit within the specification of the local service.

One example of local blocking or routing rules tied to a specific local service is known as a *gatherer/distributor* block. Typically, such blocks use a description that on the surface appears to be a single block to actually represent multiple blocks. To give an example, suppose a block has origin A, destination F, and traffic destinations B, C, D, E, and F, but also has a special flag set to indicate it is a gatherer/distributor block. The blocking system recognizes this flag and implicitly treats this as 10 blocks: The A–B, A–C, A–D, A–E and A–F distributor blocks (called that because cars are distributed from A to locations B–F, which are usually considered to be customer locations) and A–F, B–F, C–F, D–F and E–F gatherer blocks that pickup local traffic and gather it for further processing at F. It is possible this block has the same origin as destination. In practice, the train that carries the block may only stop at a subset of the traffic destinations, and the set of implicit block locations is the intersection of the traffic destination range and the train-route locations.

There is a second form of local service specification that gives more details on where within a customer location cars should be placed, typically at the zone-track-spot level. A zone-track-spot is a way to specify a specific siding at a specific customer, in effect a form of detailed addressing of locations within a larger station. An example is an automobile manufacturer that has specific locations for auto parts separate from locations for multilevel auto racks, but all of which is considered one station by the railroad's systems. Another example would be a mine that has some tracks in the same yard for various chemicals needed in the production of the bulk product and some tracks for loading the bulk product. The difference between this type of local block and the gatherer/distributor types of local blocks is that the zone-track-spot level blocks are usually describing movement within a station, while the more general local blocks describe movements between stations.

5.3 The Table-Based Blocking Systems OR Challenge

Given that most of the production blocking systems used by railways are table-based, and most OR-based approaches do not work well with tables, significant challenges arise in the creation of practical analytic and OR-tools to support blocking plan design. In short, while new plans can be developed using OR methods, these plans cannot be readily translated into a form usable by production blocking systems.

This situation gives rise to the need for several different types of algorithms and analytic techniques. These include ways to:

1. Translate algorithmic solutions into table-based solutions.
2. Develop incrementally focused optimization techniques.
3. Determine the quality of the current car routings in the existing table-based blocking rules, and suggest improvements.

To understand each of these issues, we must first understand in more detail the challenges of managing table-based approaches. Table-based blocking systems are easy to understand in concept, and work well in the sense that you get exactly what you specify in terms of car-to-block assignments. However, over time the rules can get very complex, and problems can arise.

One series of problems is that most changes to the blocking plan are based primarily on manual observations, and the personal knowledge of the operating plan by the persons making the changes:

- These observations tend to be localized in nature, which means that they often miss the network effect of changes. Because the system does not take into account the network impacts of a change, all changes by their very nature tend to represent local modifications to the blocking plan, unless the planner has a bigger picture perspective and acts upon that in a complete and thorough manner. As an example, the blocking tables that feed one yard can be modified independently to redirect traffic away from that yard to other system locations. This type of change may solve a local problem, but can overload the system at other locations or lead to inefficient routings. The results of this myopic view, when compared to a network-based view, are an increase in car-miles traveled, additional handlings, and potential delays due to unforeseen congestion incurred when transporting the current and rerouted traffic.
- The manual process can be efficient for small changes, but can also be very time intensive for large changes, which can manifest itself through slower response times to both planned and unplanned network disruptions. For example, if a line experiences an unplanned service outage, the planner must:
  - Identify all blocks that are affected by the disruption.
  - Manually identify acceptable reroutes for the affected traffic.
  - Manually enter the reroutes into the system by changing numerous rules at multiple locations.
  - Review the changes to ensure that (1) the reroutes are entered correctly, and (2) that the reroutes have the desired results.
- The overall process is by its nature very dependent on the skill level of the planner and can easily result in incomplete changes being made. For example, a complete job of introducing a new block requires that changes be made in not just the yard where the block is being added, but also at upstream yards. One must consider all of the traffic at the yard where the block is being added, and make sure that the rules for all of this traffic are changed, requiring many separate edits. Furthermore, by introducing a new block, it may make sense to reroute traffic to the yard where the block is being added, which requires both the vision to identify this traffic (which is non-obvious), and the need to change the blocking at various "up stream" yards to redirect this traffic. Finally, the downstream routes/blocking should be checked to make sure that the routes for the redirected traffic are efficient ones all of the way to destination.

One proposed solution, which can address many of these issues, is to replace the table-based blocking system with an algorithmic blocking strategy. This is what was done by Norfolk Southern and Canadian Pacific, and is under consideration at other railroads.

5.4 Algorithmic Blocking

The fundamental foundation of algorithmic blocking is the notion that given a set of blocks, one can find a shipment routing by finding a weighted shortest path across a network formed from the blocks. In this network, each block represents a link going from one yard to another. The cost associated with each block is generally a function of the end point yards, the physical lines traversed, the type of block, and the type of traffic being handled. The goal of the cost function is often not to represent true handling costs, but instead achieve an outcome that is consistent with current operating practices. For example, yard costs are often reflective of the bias of a railroad to use hump yards in preference to flat yards, and larger yards in preference to smaller yards.

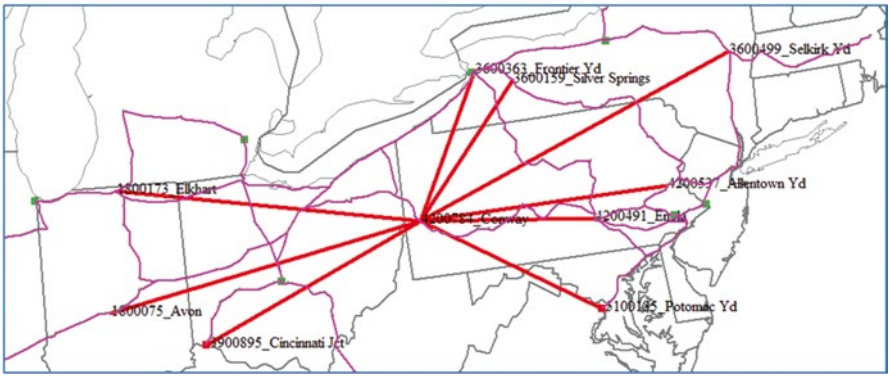
Once the blocks are defined, and the costs for each block are determined, a network can be created where the nodes are the starting and ending points of each block, and the links or arcs are the blocks themselves. By using a simple shortest path, we can then quickly determine the lowest cost routing for each shipment over a particular set of blocks.

There are still rules in an algorithmic blocking system. These rules are typically broken into two types, which are sometimes called "absolute" and "permissive." An absolute rule acts much like the rules in a table-based system and specifies that the specific cars matching the rule must be placed in a specific block. Permissive rules simply specify which blocks could be used by a shipment, and are used to develop a list of potential or candidate blocks for moving a shipment. These rules also help to dictate the cost of using a particular block. For example, one might designate a block as being an intermodal block. If the shipment the system is trying to route using the algorithm is an intermodal car, then this block would be eligible for consideration. If the shipment was a general merchandise shipment, this same block would not be considered by the algorithm. In this way, the user can control the choices available to the algorithm when it selects blocks to move a particular shipment.

Both the absolute rules and the permissive rules are based on matching shipment attributes to the attributes associated with each rule. There are no limitations to the types of attributes that can be used, and these attributes are generally the same as those described above for the table-based system.

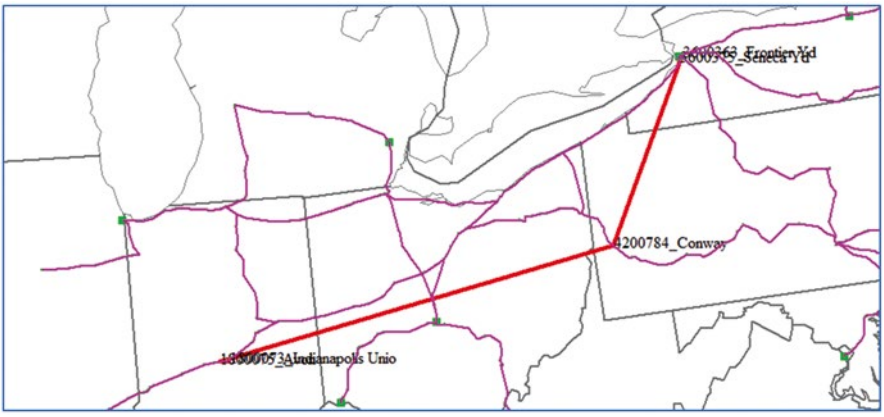
The most difficult part of defining a blocking plan is usually the specification of the car-to-block assignments through tables. The algorithm simplifies this tedious step, saving significant amounts of time and allowing large-scale blocking plan changes to be implemented more quickly and accurately than in a table-based system.

To understand the concept of a "network of blocks," please see the following diagram that shows all of the outbound blocks made from one location:



In the algorithmic process, each of these blocks is considered a network link. By combining this single location view with the blocks made at all other locations, we can create a complete network of blocking options. The costs of each link (or block) in the network is determined by the attributes of the shipment, the yard where the block is made, and various user controlled parameters. Only those blocks that are eligible to carry the car being routed are considered in this process. Where an absolute rule exists for a particular car at a particular yard, only one link would emanate from that yard.

Once formed, the algorithm can find a complete, end-to-end solution for routing a car across the blocks. Such a solution might look something like the illustration below that shows (although some details are hard to see) a traffic record from Indianapolis, IN to Seneca, NY taking four blocks: First is a local block to "Avon yard" near Indianapolis, second a block to Conway yard near Pittsburg, PA, third a block to Frontier Yard near Buffalo, and fourth a local block to Seneca yard.



In order to work, the blocking algorithm must know the destination of each block. This is a major difference when compared to the table-based blocking systems, and can be used to provide guidance to the trip planning system with respect to the validity of block-to-train assignments and the location of block swaps.

The real strength of this algorithm-based process is the ability to assign cars to blocks with a significantly reduced need for large tables specifying which cars are to travel in which blocks. When major changes need to be made in a table-based environment, the editing of the tables with their thousands of entries becomes a major barrier to the ability to undertake the change. Furthermore, in the planning environment such tables are a barrier to examining the full breadth of options available. In addition, the table-based approach is strongly influenced by the skill level and care taken by the analyst. While the algorithm-based approach cannot guarantee a specific outcome for each shipment being routed, it does assure that cars are routed consistently and efficiently. Furthermore, by adding in "absolute rules" the algorithm can be forced to produce specific outcomes when necessary.

Norfolk Southern and Canadian Pacific are the only railroads known to the authors to be using an algorithmic-based approach in their production systems. A number of railroads use algorithmic approaches in the planning process and to support optimization. At this time, to the best of the author's knowledge, the translation of the algorithmic planning and optimization results into table-based solutions for use in the railroad production systems is a largely manual process.

5.5 Examples of Areas Presenting OR Challenges

The issues and analytic needs related to the blocking plan can be divided into several sub-problems or topics:

- *Blocking plan design*—typically an offline process that results in either incremental plan changes on a daily or weekly basis, or more sweeping changes on a more periodic basis.
- *Specialized blocking situations*—due to a combination of factors, there are many specialized situations that must be addressed by blocking systems. Examples include the need to specify blocking between railroads, separate service for different types of customers and lines of business, the need to specify how local services will be produced, and the management of capacities. When applying OR techniques, one often simplifies the problem in order to generate feasible solutions within acceptable computational limits. However, these simplifications often mean that either only a subset of the business can be modeled or optimized, or the solutions require significant manual adjustments to permit their use for actual operations. This is a significant ongoing limitation for most algorithmic and optimization-based blocking tools.
- *Blocking plan optimization*—a number of optimization methods have been developed, and applied with varying degrees of success (Ahuja et al. 2007; Van Dyke 1986, 1988; Bodin et al. 1980; Barnhart et al. 2000; Newton et al. 1998;

Newton 1996; Crainic et al. 1984; Gorman 1995; Keaton 1989, 1992; Yaghini et al. 2012). In addition to the citations provided, a number of organizations have successfully developed and deployed their own optimization models such as Norfolk Southern, CSX Transportation, and Oliver Wyman. The currently available techniques have a number of limitations. Generally, they are only for a single line of business (carload, intermodal, etc.), and they generally can only handle a subset of the problem. In particular, most of the existing methods do not do a very good job of handling the design of local blocking plans, and tend to only support generic carload blocks, and thus do not take into account a variety of special situations. Additional challenges arise when it comes time to adopt the solution, particularly with respect to translating the optimizer results into a set of table-based blocking rules. In the author's experience, this results in a strong need to manually review optimization results, and for using experienced planners to pick and choose from the optimization results those blocking changes that should be implemented. It also means that using an incremental approach to blocking plan improvement may be more effective than a "clean sheet" type approach.

- *Dynamic blocking concepts (time-based blocking)*—most current blocking systems focus on either use of rules-based routing, or algorithms that minimize a combination of distance traveled and handlings incurred. A number of time-based strategies are also possible (Kraft 1998), and some are being explored by a few railroads such as Norfolk Southern Corporation (INFORMS 2010) and BNSF Railway. These include both continued use of fixed routings of shipments, and also dynamic routing strategies. Both the dynamic routing approaches and fixed routing strategies are explored in Chap. 4. The fixed routing strategies attempt to factor in transit time into the static cost formulas along with distance and handlings. These static routing strategies try to reflect the available trains to move each block, and in some cases may reveal that a different routing for some shipments may produce better results in terms of total transit time, with no appreciable change in the other cost factors. As noted above, the routings for shipments are still fixed, but time will thus be added to the decision factors. Under dynamic routing strategies, the current status of each train relative to its carrying capacity is taken into account in deciding which sequence of trains and train-blocks should be used to advance the shipment on a real-time basis. These approaches are being explored, and used to some extent by at least two North American Class I railroads. In both cases the approach leverages a time–space variant of the existing shortest path algorithms described in this chapter.
- *Block-to-train assignment*—blocks are carried by trains from their origins to their destinations. Typically, the train design problem is treated separately from the block design problem for both historic and problem complexity reasons. This topic is primarily addressed in detail in Chap. 1 on train scheduling.
- *Execution support*—the systems and processes that are used in real-time to assign shipments to blocks and route the shipments to destination could benefit from stronger analytics and decision support to help guide real-time management decisions. This topic is examined in Chap. 4.

5.6 Semi-manual Blocking Plan Design Techniques

This section describes various techniques used to develop blocking plans. In practice, most blocking plans are developed incrementally, however this section will review both incremental and clean-slate approaches that have been done in practice. This section is devoted to semi-manual techniques that have been used routinely by railways; we delay until Sect. 5.8 the discussion of automated blocking plan design techniques using optimization methods.

5.6.1 Incremental Blocking Plan Design Techniques

One of the most common activities is to identify incremental changes to an existing plan that will improve overall performance of the plan, or address specific issues such as keeping workloads at a specific yard within its capacity limits. Thus, in these techniques we assume that there is already an existing blocking plan.

When determining what type of incremental changes to make, the first consideration is to establish the strategic reasons for making a change. Two examples are:

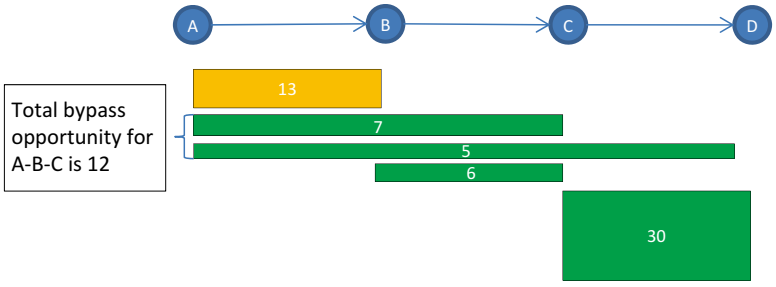
1. *Tuning an existing plan*: Traffic volumes have changed from when the existing blocking plan was created, and there is a need to fine tune the plan to better match the new levels of traffic.
2. *Change traffic volume at a yard*: There are various reasons that planners often have for wishing to increase or reduce the classifications per day made at a yard. A frequent reason is when traffic volumes change, a yard might become overloaded. In some cases, even a seemingly small change such as increasing from 2,000 classifications per day on average to 2,050 per day, proves to be a tipping point causing instability at the yard. Less common adjustments are during studies of yard expansions, where capacity is increased and the cost per classification at the yard declines. One part of the overall analysis on the cost/benefits to expanding the yard is to understand how the changes affect the overall network.

5.6.2 Tuning an Existing Plan

Two of the most common tools used in undertaking incremental, manual tuning of a blocking plan design are (1) the identification of bypass opportunities and (2) reviewing low-volume blocks.

A bypass opportunity occurs when two blocks in a sequence carry many of the same cars. For example, if block A–B carries 25 cars/day and block B–C carries 18 cars/day, and of these cars, there are 12 cars/day that travel on the A–B block and then the B–C block, we have a bypass opportunity of creating a block A–C to carry those 12 cars/day.

There can also be bypass opportunities that span several blocks, and in many cases the various bypasses can compete with each other for use of the same traffic. This is illustrated in the diagram below, where there are 13 cars/day that use the A–B block but not the B–C block, 7 cars/day that use the A–B–C sequence but not the C–D block, 5 cars per day that use the A–B–C–D sequence, 6 cars per day that use the B–C but not the A–B or the C–D blocks, and 30 cars/day that use the C–D but not the B–C block. This results in several bypass opportunities, one for 12 cars/day by creating an A–C block, one for 5 cars/day by creating an A–D block. However, if the A–D block is created, along with the A–C block, the A–C block would have only 7 cars/day instead of 12 cars/day.



Usually, a list of all the bypass opportunities is created, and filtered for larger opportunities and sorted either by cars per day or car-miles/kilometers per day. Further filtering is done to eliminate artificial opportunities caused by interchange and local blocks. Interchange received blocks are often a source of bypass opportunities, however to implement these requires the cooperation of the delivering railroad, which can require a significant commercial negotiation. For this reason, interchange bypasses are often handled as special cases or skipped in most analyses. Likewise, many local block bypass opportunities are not feasible due to operational and yard capacity constraints, and thus must also be ignored in the bypass analysis process.

Calculating bypass opportunities creates indications of where potential new blocks could or should be made. However, the analysis is only indicative. It takes further analysis before the planner can actually add a new block. For example, the planner needs to decide if yard A has the capacity to make an additional block. If not, the planner might need to eliminate a low-volume block that originates from A. Adding a bypass block may also substantially reduce the volumes of other blocks. Therefore the planner needs to be judicious in the application of bypass blocks, and generally the planner makes only a few changes before flowing traffic over the revised plan to obtain more precise estimates of block volumes.

Finding bypass blocks is the same whether the underlying blocking plan is algorithmic or table-based. However, implementing bypass blocks in a table-based plan is notably harder since the rules need to be setup to attach the right traffic to the new block. When the rules are mostly traffic destination based, then the yard relaxation algorithm (described in more detail later in Sect. 5.6.6) works well to automate the changes in the blocking rules related to which traffic destinations should be assigned

to the new block. This same process can also be used to identify other traffic at the yard that may want to take advantage of the new block, where this additional traffic is not currently riding on the bypassed block.

The other tool used in refining blocking plans is fairly simple—identify zero and low-volume blocks as candidates for removal. In many cases, these low volume blocks need to exist to protect service to lightly used stations. Thus, removing low, but positive volume blocks, may result in some traffic being stranded if the removed block was the only way to reach the impacted location. In table-based systems, removing a block almost certainly dictates necessary changes in the rules for other blocks originating from that yard, since the block usually is intended for specific traffic destinations. In algorithmic blocking, removing a block is much easier, although it may still result in unmoved traffic. While removing a zero volume block will not result in any stranded traffic, there may be seasonal or periodic traffic for the location that must be protected. In these cases, the zero-volume block cannot be removed.

After removing a low-volume block, there needs to be a check to see if the circuity for the new traffic routings is too large.

5.6.3 Checking Circuity and Excessive Handlings

Two other techniques that are employed in developing a good blocking design are circuity and excessive handling analysis. Circuity analysis computes, for each traffic record, the ratio of the distance of the traffic route as given by the block sequence to the distance of the shortest path between the origin and destination of the traffic record. Traffic with large circuity (often thought of as being 1.2 or larger) is studied to see if changes in the blocking rules or adding a block are warranted. The studies usually take into account the volume of traffic, ignoring very small flows of high-circuity traffic.

Excessive handlings is the analysis of the number of classifications a shipment undergoes. As a rule of thumb, high-volume traffic with four or five intermediate classifications are usually examined to see if additional blocks or blocking rule changes are necessary.

When adding blocks to solve either circuity or excessive handlings issues, the planner usually looks at existing blocks to see if there are any simple reroute options that would solve the problem. These reroutes can be identified using the relaxation techniques described in Sect. 5.6.6, or in some cases through bypass analysis.

5.6.4 Change Traffic Volume at a Yard

Tools used for adjusting volume (workload) at a yard differ between algorithmic and table-based systems. Usually the planner has a target traffic volume in mind that should get added or taken away from a yard. With blocking plans that are

table-based, the planner tries to identify a set of traffic whose expected volume is close to the target and then creates specific rules that address that set of traffic.

Unfortunately, the result of repeated tuning of the volumes through the yards can lead to complex rules, underutilized blocks and costly routings in some cases. The general idea is to go to "upstream" locations that feed traffic into a yard, and change the traffic routings for selected traffic from these upstream locations to use alternate yards than the one with an excessive workload. Of course, in making these adjustments, the impact on the yards to which the traffic is redirected must also be assessed, and the resulting circuity and handling levels must be assessed.

With algorithmic routing, planners can change the penalty (cost) of classification at the yard with too much traffic to influence the amount of traffic being switched at the yard. However, that has two problems:

- The changes in traffic volumes are often a step function with respect to the classification cost. For example, as the cost increases, the traffic stays flat for a while, then has a "jump" down and then stays flat until the next jump occurs. This is caused by groups of traffic changing their routings as various tipping points are reached in the relative cost of using the target yard compared to other yards. In some cases it can actually be hard to find the right cost parameter to use, and due to the step function there may be no "right" cost parameter.
- Planners are often loathe to change the classification cost because it might change many routings that they do not want to change.

Because of these two reasons, typically planners using algorithmic blocking systems use the same techniques as those using table-based blocking systems, which is to change the rules on the blocks at the upstream yards in a precise manner. However, they may use changes in the yard's penalty or cost as a way of identifying potential candidates for reroute.

5.6.5 Designing Blocking Plans Using a Clean-Sheet Approach

In the previous section, we discussed the use of bypass opportunities and removal of zero or low-volume blocks to take an existing blocking plan and improve it. To start without an initial blocking plan (variously called a clean-sheet, greenfield, cold start or zero-based approach), the usual practice is to generate a simple, but largely feasible initial block plan based on a yard hierarchy, and then incrementally improve it as discussed above. Usually at this step only algorithmic blocking is used, generally with blocks that have no specific traffic rules (but may be focused on specific lines of business). This allows a good plan to be formulated more quickly. Additional explanation of building an initial blocking plan and optimizing a plan is given in Sect. 5.8 below. Often in clean-sheet approaches, the local plan is not changed, and only the longer distance blocks are examined. Furthermore, interchange blocks to and from other railroads may be kept frozen, as these can only be changed through bilateral negotiations with each of the connecting railroads.

In a typical manual approach the steps are as follows:

1. Create an initial block plan. There are two strategies:
   (a) Take the existing blocking plan, and remove all of the non-local and non-interchange blocks, or
   (b) Use a hierarchical approach to generate a starting set of carload blocks that cover the movement of all of the traffic (see Sect. 5.8 below). This plan typically connects local or serving yards to larger regional and system yards on a nearest neighbor basis, and then connects regional yards to system yards, and system yards to each other.
2. Either exclude the non-carload traffic from this process, or use specialized logic to create initial blocks for other lines of business (unit train, intermodal, automotive, and grain).
3. Flow the traffic over this starting set of blocks using an algorithmic approach.
4. Review the plan, looking at circuity analysis, excessive handlings analysis, and bypass analysis to identify where new blocks could be added, and low volume analysis to identify blocks that might need to be eliminated. Examine yard volumes and the number of blocks made at each yard, and identify where adjustments to penalty costs to bring these yards into conformance with their capacities might be made.
5. Make some of the changes identified in step 4 above, and flow the traffic over the revised blocking plan.
6. Repeat steps 4 and 5, monitoring various key performance indicators (such as handlings, car miles, yard volumes) until the plan is satisfactory.
7. As appropriate, review and revise the local and interchange blocks as a separate review exercise.

The above process can be automated through an optimization approach, which is discussed in Sect. 5.8 below. Using a manual approach, it is the author's experience that a well-trained team can complete the steps above in a 1- to 2-week period for a large railroad. While optimization can produce an initial plan in a few days (including setup time), we have found that the resulting plan must still be manually reviewed and refined using some of the above steps in order for the plan to be acceptable to railroad management.

5.6.6 Tuning Table-Based, Traffic Destination Attribute Rules Using Relaxation

As indicated in Sect. 5.2, the most common type of table-based rule is the set of allowed traffic destinations for a block. Due to the manual nature of maintaining table-based rules, sometimes individual traffic movements are routed inefficiently and at higher cost than necessary due to using a less than optimal set of traffic destination rules on the blocks. This typically happens when the blocking plan is

changed, but not every destination assignment is updated to reflect the change. This can happen when blocks are added or removed, or when rules are introduced to change the amount of traffic handled at a yard resulting in some traffic being purposefully routed non-optimally. Later there may be no need to route the traffic that way, but the change is forgotten and not undone.

A technique, *called rule relaxation*, can be applied to discover cases of non-optimal routing. Rule relaxation applies algorithmic-based routing to a set of existing blocks by ignoring traffic destination-based rules. This is easiest done for railways that use two levels of rules—where one level is based on all the blocking attributes and is placed higher in the rule priority order, and the other level is based strictly on traffic destinations. The concept is that algorithmic routing will find the least costly block sequence.

The broad-based approach to rules relaxation applies this approach to a large set of blocks and traffic at a network level. For example, we might apply this to all general carload traffic. All of the blocks in the table based plan would be reviewed using computer-based business logic, and each carload block identified. The carload traffic would then be flowed across the blocks using a modified algorithmic approach. Any complex rule would be retained, and applied to the traffic on an absolute basis. But for traffic not hitting such rules, an algorithmic approach would be used to flow the traffic using the carload blocks identified by the business logic. This will result in more optimized routings for some of the carload traffic, where the changes can be identified through a comparison to the routings produced by the pure table based approach (possibly using the triplet analysis described below).

This approach, while powerful, can be difficult to use. It can result in a large number of routing changes, which then need to be reviewed by the planners. Experience has shown that many of these changes are either of trivial value or unacceptable for operational reasons. The result is that the cost/benefit of this approach can be perceived as negative, or the process can simply overwhelm the planer. Furthermore there can be no automatic adjustment of the rules based on this kind of analysis due to the risks of unintended consequences, which means that the changes must be manually entered into the rules tables.

To understand the reason that more complex rules may need to be retained in the relaxation process, consider a yard with four blocks A–B, A–C, A–D, and A–E. The rules for the blocks are prioritized as follows:

1. If commodity is hazardous and traffic destinations are X, Y or Z, take block A–B.
2. If traffic destinations are X, Y or Z, take block A–C.
3. If commodity is hazardous and traffic destinations are P, Q or R, take block A–D.
4. If traffic destinations are P, Q or R, take block A–E.

The A–B block is for hazardous materials going to X, Y or Z, and based on the rule ordering the A–C block will implicitly be for non-hazardous material for traffic destinations X, Y and Z. Let us assume that going to C is a better, cheaper route for traffic destinations X, Y or Z compared to going to B. Under an open relaxation schema that discards the more complex rules, these four blocks will no longer be ranked relative to each other, because the rules will now be permissive. Now block

A–C would be able to take both non-hazardous and hazardous material, and since going to C directly is less expensive for traffic destinations X, Y or Z, it will naturally attract all traffic going to those locations. This illustrates the pitfalls of using completely open relaxation.

There is a second form of relaxation, called *yard relaxation*, for table-based blocking plans that examines and recommends traffic destination rules at a single yard. It is simple, but is well received by the planners and is considered quite valuable. It is primarily for railways that use traffic destinations as the primary blocking rule attribute. It also has the advantage of limiting the amount of information that needs to be reviewed by the planner, making it a much more understandable and approachable way to improve the plan.

In simple terms this is an exhaustive search approach. To work, one selects a specific yard (which we will call yard A). One then takes some set of candidate traffic movements and tests how each traffic movement would currently be routed from yard A, and how the traffic would perform if it used each of the other blocks that are made at yard A. The cost of the current routing is compared to each alternative, and the cases where improvements are realized by changing the routing are identified. As with other forms of relaxation, care must be taken not to test inappropriate cases such as putting an intermodal movement on a coal block. To protect against this, such relaxations are often limited to only carload traffic and only carload eligible blocks made at yard A are tested.

This approach can use either the existing traffic at yard A as the candidate traffic, or can generate a set of candidate traffic movements. The generated movements can have the advantage of providing test cases for all possible destinations on the railroad, supporting a more thorough review of the routing rules. In the generated case the user typically specifies a standard profile for the traffic movement attributes, such as a generic, loaded boxcar carrying a common, non-hazardous commodity.

In more mathematical terms, the process can be expressed as follows:

1. Let the yard in question be called A, and let the destinations of the blocks that originate from A be $BD = \{B, C, D, ..., H\}$. As noted above the set BD might be limited to only the carload blocks.
2. Execute a double loop—first for each possible candidate traffic destination (say d) then for each possible block destination in the set BD (say r).
3. Find the block sequence from r to d, and add to this the cost to go from A to r.
4. After going through each r in BD, find the block destination r^* whose cost from A to r, then r to d, is the smallest.
5. Assign the traffic destination d to block r^*, and repeat for the other possible traffic destinations.

At the end of the process, each block that originates from A has a set of preferred traffic destinations that can be compared to the current routings.

When the yard relaxation process suggests moving a traffic destination from one block to another, the process should calculate various metrics such as the total distance and total block sequence cost. This gives the user the ability to examine the proposed traffic destination assignment and make changes to the rules if necessary, potentially ignoring proposed changes that have only a small benefit.

5.6.7 Additional Methods for Testing Plans

In the previous section, two plan testing concepts were mentioned: circuity analysis and excessive handlings. In this section, we discuss other tests that a good blocking design should pass.

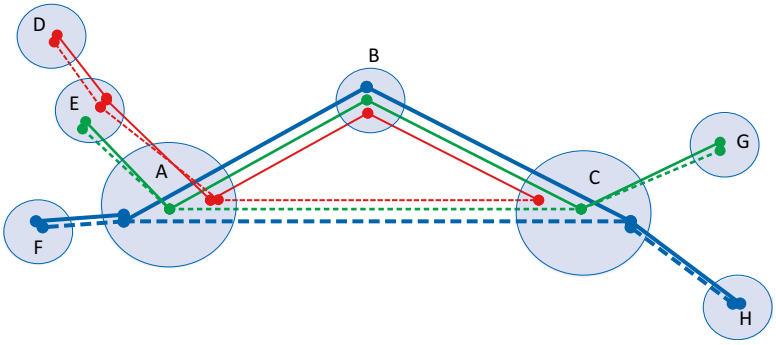
- *Unmoved traffic.* Most traffic—typically at least 98 % of the volume should have block sequences or routings. The large railways may have some traffic that cannot get a block sequence because no local block is defined either from an origin or to a destination, however operationally they usually know how to move this traffic should it occur.
- *Completeness.* The notion of completeness is that a blocking plan should be able to move any possible traffic that could at some point be tendered to the railroad. Even if all the existing or known traffic is moved, there is no guarantee that all elements of a future set of traffic records could be moved since the traffic set that is used for analysis (usually based on historical data) does not have all the combinations of origins, destinations and various other attributes that can arise. Railways sometimes have a "test traffic set" that they use for testing the completeness of the plan. It is generally composed of all origin/destination pairs for a generic shipment such as a standard, loaded box car, as well as all reasonable intermodal and automotive origin/destination pairs for the appropriate car types. This test traffic set may also have other records for specialty traffic cases.
- *Loops.* A common error in table-based blocking systems involves plans that generate block sequences with loops. For example, a traffic record from A to E might first take the A–B block, then the B–C block, then the C–D block, then a D–B block. At this point, the block sequence loops and keeps cycling. This would occur if, say, the D–E block has a lower priority than the D–B block and both could accept a traffic destination of E. Most block sequencing or routing procedures contain a test for loops, terminating the routing of individual shipments when loops are detected. Broader-based testing for loops can be done using the same "test traffic set" as is used for completeness testing (though loops are often caused by specialized rules for specific subsets of traffic).

5.6.8 Triplet Analysis for Blocking Plan Comparisons

Triplet Analysis is used to compare the block sequences from two different blocking plans for the same traffic set to understand fundamental routing differences between the plans. It works by examining the block sequence for each traffic record in the sample that has a different sequence between the two plans. Because there can be many individual traffic records that share the same routing difference, looking at individual traffic records can be time consuming and make it hard to identify the larger patterns. Triplet analysis attempts to identify the underlying patterns across multiple traffic records.

Triplet analysis has two primary values: It dissects and ranks the differences between two blocking plans, and it allows a user to examine selected differences and pick which ones to use. This is particularly important for using the current block optimization technology that cannot, by itself, produce a complete realizable plan but together with triplet analysis can produce useful modifications to an existing plan.

Consider traffic D–C, E–G, and F–H. Suppose in plan 1, the block sequences were (respectively) D–E–A–B–C, E–A–B–C–G, and F–A–B–C–H and in plan 2 the block sequences where D–E–A–C, E–A–C–G, and F–A–C–H. This is illustrated below, with the original block sequences in solid lines and the new ones with dashed lines.



All three traffic records have a sub-sequence of A–B–C in plan 1 and A–C in plan 2. Typically these routing differences occur for the same reason, hence grouping together these traffic records for analysis can be used to generate statistics that highlight the underlying differences between the two plans.

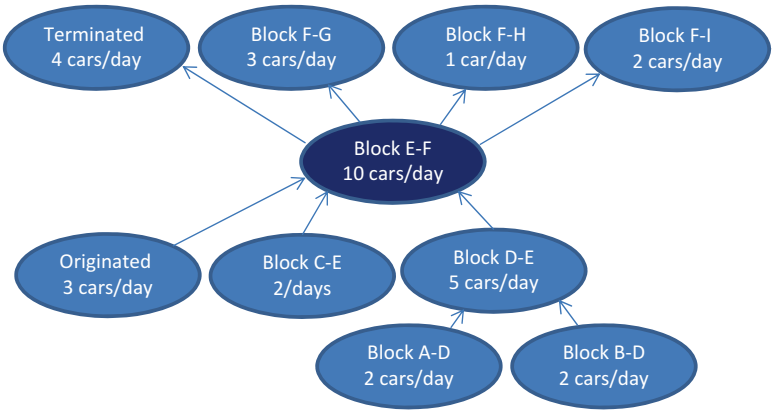
Triplet analysis has three components:

1. Identification of routing differences, as illustrated above.
2. Calculation of business statistics for each triplet such as the distance of the plan 1 route versus the plan 2 route, the car-miles/kilometers, the number of intermediate classifications, and the equivalent car-miles/kilometers when the intermediate classifications are converted to an equivalent distance metric. These statistics are critical to the ranking of the triplets.
3. Identification of which blocks occur only in plan 1, which blocks occur only in plan 2, and which ones are common. For example, if all three blocks (A–B, B–C, and A–C) are common, it suggests that the difference between the two plans is due to block-rule changes or change in the cost of classification at B.

It is called “triplet analysis” because the canonical example used when explaining how it works is the above example that involved three blocks, and in practice that is a common situation. Four blocks also commonly occurs (one route may be A–B, B–C, and the competing route is A–Z, Z–C) and sometimes more than four blocks.

5.6.9 *Tree View Analysis*

Tree views show how traffic flows through downstream and upstream blocks. Graphically, it is represented as illustrated below for the block E–F. It shows how the volume of block E–F flows from and into other blocks for a specified depth. The block D–E may have 20 cars/day, but only 5 cars/day subsequently go onto block E–F. Originations and terminations are usually shown for the E–F block and not the upstream or downstream blocks.



5.7 Specialized Blocking Situations

A number of specialized situations need to be factored into any blocking plan solution. These are discussed below.

- *Local Services*: Every railroad has a unique approach to the specification of local blocking. Many of the issues related to local services are discussed in Chap. 4, and to some extent in Sect. 5.2 above. In general, the most detailed rules in the blocking system are related to local blocking, and in some cases local blocking can represent 50 % or more of all the blocking rules. Both table-based blocking systems and algorithmic blocking systems require extensive rules to specify the local blocking because of the high degree of detail required to get shipments to the right customers on the right tracks. Furthermore, because many local block assignments occur at the destination of a trip, the algorithmic systems generally use a process similar to that used by the table-based systems for assigning the final yard-block code. At some railroads, the local delivery rules are maintained not in the blocking system, but in some kind of local service specification process that can be part of a separate local services database or part of the main train database. In these situations, the blocking system must either look at this external

data source to determine the local blocking, or the blocking system must receive periodic rule updates that are derived from this external data source. From an optimization perspective, the local blocking is often treated as fixed, and the optimization focuses on the longer haul elements of the blocking plan.

- *Interchange blocks/run-through trains*: In some cases one railroad agrees to build blocks that "interchange" or go onto another railroad. For example, railroad A might agree to make five blocks for railroad B for interchange or handover at a specific location (or junction) in exchange for railroad B making five blocks for railroad A. Typically, the receiving railroad sends instructions to the delivering railroad that specify which shipments should go into each block. When these blocks are placed on a train that does not stop or get broken apart at the interchange, one gets a "run-through train." The biggest difficulties with interchange blocks and run-through trains are in maintaining accurate rules for assigning shipments to the right interchange blocks, and knowing in advance which block each shipment will be in when received from interchange. Problems with both issues can contribute to the generation of inaccurate initial trip plans on a real-time basis, as well as represent a challenge to the planning/modeling process. Optimization routines often get into trouble when they change the interchange blocks relative to the existing agreements with the other railroad in terms of either the number of blocks made or their content. From a blocking system perspective there are four things to consider:

  i. A back-up set of tables must be maintained for each pre-block the railroad makes that roughly mirrors the instructions that would otherwise be received from the foreign road (these instructions come through a data interchange process in North America known as the 419/420 message exchange process). These back-up tables ensure the continued functioning of the classification process in cases where the communications protocols fail, and support the modeling/planning process.
  ii. Each railway must maintain a definition for each pre-block it will ask a foreign railway to make, both to support the 419/420 process, and to provide back-up instructions for entry into the foreign road's computer systems.
  iii. Many railroads embed a special code on the waybill or movement record that reflects the pre-block assignment for both interchange received and interchange delivered traffic, often based on the 419/420 process, that can be seen and used by the blocking engine when doing its car-to-block assignments.
  iv. The railroad has the freedom to make pre-blocks in multiple locations on the railroad and the blocking system should be adjustable enough to optimize where traffic is classified for these blocks.

- *Handling of specific specialized services*: Most blocking systems are focused primarily on conventional carload traffic. Blocking can be entered into the system for a variety of specialized traffic such as grain, automotive, intermodal, and unit trains. Some services, such as unit coal traffic, do not fit very well to the

traditional car scheduling and blocking paradigm. Others such as grain sometimes fall within the scope of the trip planning and blocking process, and in other cases do not. Finally, some services such as intermodal come pretty close to the trip planning/blocking process described above, but have complexities related to there not being a one-to-one correspondence between the railcars and the materials (boxes) being shipped. The result is a need for either special logic within the trip planning and blocking environments for each of these cases, or the exclusion of this traffic from the classical trip planning and blocking processes. The degree to which this can be done with any accuracy depends on a combination of business logic, data quality, and the way the blocking plan is designed. The degree of use varies widely for traffic such as grain and intermodal. The handling of these special cases is discussed in more detail later in this chapter.

- *Multi-location yards*: The concept of a multi-location yard is mostly specific to the rules-based blocking systems. Each set of rules is location-based. If one has many locations that have blocking rules, this can result in a huge number of rule sets and rules to maintain. In some cases there can be clusters of locations that will have similar or identical blocking. For example, a series of local yards all served by the same trains might have essentially identical blocking. To reduce the number of location-based rule sets and ease the manual maintenance process, the concept of the multi-location yard was developed. This is a situation where a set of locations all use the same set of blocking rules. To the extent there are differences between the locations, rules are created that use a "current location" attribute to restrict their applicability to only one location. These multi-location yards can increase the complexity of the blocking business logic in some cases, and there are indications that railroads are moving away from this concept. The multi-location yard concept is not used by algorithmic blocking systems such as the one used by Norfolk Southern and Canadian Pacific Railway.
- *Data clean-up issues*: A number of data sources are used as inputs to the blocking process, and many of them can have data issues associated with them. As a result, the blocking processes have a variety of mechanisms for correcting these data issues in order to improve performance. These include the ability to specify "variants" of spellings in the blocking rules, and "waybill correction" tables to standardize shipment attributes prior to their use by the blocking system.
- *Block assignment regeneration*: From the blocking plan generator's perspective, there is no specific requirement to "regenerate" shipment-to-block assignments. The blocking system simply processes requests when it is given the current location of the shipment, the targeted destination, and a set of attributes to be used in determining the appropriate shipment-to-block assignment. Based on that it provides back a block. External monitoring systems, including yard systems and trip planning systems, are responsible for determining when the shipment-to-block assignments need to be requested or regenerated.
- *Capacities*: In general, capacities of individual blocks are not considered in the design and maintenance of the blocking plan. In some cases capacities are considered during real-time execution of the plan with respect to specific trains and/or yards.

This subject is addressed in Chap. 4. During the design of a blocking plan, minimum volumes for individual blocks are often used to measure plan quality and as constraints on block formation. Maximum capacities of yards to handle railcars and maximum numbers of blocks that can be made at a yard are often considered during plan design and optimization. This use of capacities is explored further later in this chapter.

- *Hold Blocks*: Hold blocks are an important concept used to classify or sort a set of shipments into a group that does not have an outbound train, and thus requires manual intervention in the handling of the shipments. Essentially they can be viewed as a forced blocking plan failure. These hold blocks are used for a variety of purposes, but the most common is to collect shipments that will require manual intervention prior to onward movement. Common usages are for grain that may be assembled into solid or unit trains for onward movement, and for the collection of empty railcars. Hold blocks pose a particular challenge for algorithmic blocking systems that are focused on driving shipments to their destinations. They are often modeled as "regular blocks" from the algorithm's perspective, with a flag on them that indicates that they should not allow the routing process to progress once a shipment is assigned to such a block. This can also pose problems in statistical analysis as these shipments do progress to downstream locations during actual operations (after manual handling), and stopping their forward movement in the analysis process tends to understate railcar distance and handlings at yards.
- *Alternate Destinations*: In some cases the destination of a shipment can change depending on how it is routed. Four common examples of this are:

  i. Situations where a shipment is placed into "constructive placement" due to the inability of a receiver to accept the shipment.
  ii. Specification of en-route stop offs, where a shipment must go "via" a specific point for partial loading or unloading, or for actions such as cleaning or en-route weighing.
  iii. Substitution of alternate destinations relative to the "billing destination" found on the waybill.
  iv. Cases where a railroad has the option of delivering a shipment to an alternate interchange point to another railroad based on operational convenience.

  Each of these cases must be handled using special logic. The first three cases are the simplest. In cases (i) and (iii), the system typically has a way of "substituting" an alternate destination based on a table of some variety. This substitution is typically handled as a pre-process to the assignment of the shipment to a block, and thus has little impact on the blocking system design. For case (ii), logic is typically added to use the "via point" as the destination for the shipment until the shipment reaches that point, and then use the final destination thereafter. The last situation (iv) is the hardest, as this represents the possibility of dynamically changing the destination based on the circumstances. In table based systems, there are typically two solutions. One is to provide some kind of look-up table that sets the targeted destination based on the current location of the shipment. As the shipment advances to each location based on the blocking, the look-up

table is consulted to see if an alternate destination should be used from that point forward. The second approach in table-based systems is to simply drive the shipment to a particular interchange point using the rules. In this situation, certain blocks are designated as "interchange blocks" and the shipment is treated as being complete whenever it is placed into an interchange block. For algorithmic systems, the standard option is to provide blocking choices to both interchanges, with a low or zero cost "phantom block" between the two interchanges that allows the shipment to reach its officially designated destination. Such phantom blocks can be somewhat challenging to specify and maintain, but appear to produce the desired result based on actual experience.

- *Re-hump Blocks*: in some cases during actual operations, the option to place a shipment into its block may not exist because no capacity is available to create the targeted block at the time the shipment is being processed, or the targeted block exists but is full. When this happens, the railcars are placed in a temporary block that will be switched into the targeted blocks at a later time. Such blocks are called "re-hump blocks" or "buffer blocks." While an important real-time operational consideration, and often needed when a yard makes more blocks than it has physical tracks, we will not address this issue in this chapter.
- *Cross-yard Blocks*: some yards are in reality compound facilities. For example, one yard complex might have separate yards for each direction, plus a local yard. These yards may not be modeled as a single location, but several separate, co-located facilities. When trains arrive, they often contain primarily shipments for a specific direction, and thus arrive at only one of these facilities. The blocking plan will then need cross-yard blocks to allow shipments to move between these facilities to reach the appropriate out-bound block. In algorithmic systems these cross-yard blocks are often set to have low costs so that such movements do not cause the yard to be avoided.
- *Directional constraints*: as noted in the discussion of the cross-yard blocks above, some yards make blocks primarily on a directional basis. This can pose a challenge during optimization of a blocking plan. If the destinations of blocks at the yard are limited to ones that are consistent with the yard's directional nature, this tends to ensure that only the most appropriate traffic is handled at the yard. To achieve this, the optimization algorithms are typically constrained to only consider the formation of out-bound blocks to appropriate locations. While one could also constrain the in-bound blocks that the yard can accept, this is often not required because the out-bound constraints will naturally limit what traffic will want to move through the yard.

5.8 Blocking Plan Optimization

In Sect. 5.6, we discuss strategies and tools that assist the planner in developing and assessing blocking plans. This section discusses strategies for automatic blocking plan optimization. A major theme in this section is that, at least at this time, there is

no technique that will generate a blocking plan that passes all the real-world constraints so that the solution can be used unchanged. However, there are two important uses of blocking optimization that give significant value:

- It provides an excellent starting pointing for developing "zero-based" (or "clean sheet") operating plans.
- When used with good comparison tools such as the triplet analysis and tree-view analysis described in Sect. 5.6, it gives planners suggestions for incremental changes to the current blocking plan.

The main reasons why the current state-of-the-art for automatic blocking plan optimization is not able to develop final, usable plans include:

- The current blocking optimization models are for a single type of block (usually for "manifest" or "merchandise") with no differentiation for types of traffic. For example, the blocking optimization techniques cannot generate a set of blocks for automobile traffic (finished vehicles or parts) plus a set of blocks that allow both auto and manifest.
- Blocking optimization does not do a good job on "local" blocking for two reasons. The first is due to the need for significant use of rules to move the car the last mile. The second is that the yard capacity constraints are more complicated for local blocks since local traffic may be moved less than daily, or it could be that several local blocks might occupy the same track and be switched just before delivery.
- Blocking optimization by itself does not make strategic changes or trade-offs. For example, the railway might decide to change the function of a yard—say eliminate the hump, or change a flat yard to focus strictly on automotive traffic.
- A complete operating plan specifies the blocking plan, the train plan, and the assignment of blocks to trains. Often when developing the train plan, adjustments need to be made to the blocking plan to reduce block swaps, circuity of the blocks or changes to ensure trains are sufficiently filled out with cars.

5.8.1 Considerations That Automated Blocking Optimization Techniques Should Consider

So far, the main characteristics we have discussed for designing a blocking plan are:

1. Find a plan that allows all traffic to have a block sequence.
2. Minimize the sum of the costs of the block sequence (the cost is a combination of the distance the cars travel and the switching costs expressed as a distance penalty).
3. Limit the number of classifications made in a yard to fit the capacity of the yard.
4. Limit the number of blocks made in the yard to fit its capacity.

However, experience has shown that these criteria alone are not sufficient, and additional constraints need to be added. These include:

1. Progressive block size. Each block is given a minimum block size, and generally the block size increases with the block distance. For example, blocks traveling less than 100 miles may have a minimum block size of 5 cars, while blocks going greater than 500 miles might get a minimum block size of 20 cars. Large long distance blocks may become "anchor" blocks and have a train that carries them from origin to destination, perhaps with some minimal circuity so that it could process some other blocks along the way. Small long distance blocks would not have a train designed around them, and often would have to be carried using one or several block swaps, and hence are not desirable.
2. Directionality of blocks. Some yards, due to the physical track characteristics and presence of other nearby yards, are often constrained to make blocks that go in only a limited number of directions.
3. Local blocking. We already mentioned that block optimization does not work well for local blocks. In our experience, it is best to "roll up" the traffic to serving yards (the second smallest tier in the hierarchy—serving yards generally handle cars for several local yards), taking the local blocking plan design out of the optimization process. Such roll-ups also have the advantage of making the problem more compact, by reducing the number of locations to be considered, the total number of blocks in the plan, and the size of the traffic database. The traffic is reduced because the roll-up process reduces the number of unique origins and destinations for the traffic, allowing similar traffic records to be combined with each other.

5.8.2 Mathematical Representation of the Block Design Optimization Problem

There have been a variety of efforts to develop railroad blocking plan and railway operating plan optimizers dating back over many years (Ahuja et al. 2007; Van Dyke 1986; Van Dyke 1988; Bodin et al. 1980; Barnhart et al. 2000; Newton et al. 1998; Newton 1996; Crainic et al. 1984; Gorman 1995; Keaton 1989, 1992; Yaghini et al. 2012). While a number of these efforts have produced quite good mathematical statements of the problems, computational constraints have limited these formulations usability to solve real world problems. The consequence is that most practical solutions use some form of heuristic that includes subsets of the optimization formulation or other concepts discussed in this chapter.

Given the above qualification, here we state the optimization problem more formally, assuming that algorithmic blocking will be used as the source for obtaining block sequences.

5.8.2.1 Data

A set of yards $Y = \{1,2,\ldots n\}$, where n is the number of yards.

There is an underlying set of links L where $l \in L$ represents a directed link. We represent the tail as $t(l) = y_1$ and the head as $h(l) = y_2$. That is, the link goes from yard y_1 to yard y_2 and represents the physical track between these two yards. Usually there is another link that goes from yard y_2 to y_1. The graph (Y, L) is typically very sparse, with the number of links typically only slightly larger than twice the number of yards. Each link has a distance. We assume that the yards are connected: that for every pair of yards $(y_1, y_2) \in Y \times Y$ there exists a connected path of links $\{l_1, l_2, \ldots l_k\}$ where $t(l_1) = y_1, h(l_k) = y_2$ and $h(l_j) = t(l_{j+1})$ for $j = 1, 2, \ldots k-1$.

The set of all possible blocks is $\mathrm{B} = Y \times Y$, that is, B is all possible arcs between yards in Y. Denote the origin of the block $b \in \mathrm{B}$ as $o(b) \in Y$ and the destination $d(b) \in Y$. Each block $b \in \mathrm{B}$ has a distance $\omega(b)$ that is composed of finding the shortest distance path in the (Y, L) graph from the origin of the block to its destination. Note that one could substitute a "weighted distance" or cost for each link that is not the same as the physical distance in order to reflect "routing preferences" on the (Y, L) graph.

For each yard $y \in Y$, let B_y be the maximum number of blocks that can originate at y, and let C_y be the maximum number of railcars that can be switched at y. Note that this is a significant simplifying assumption in that the maximum number of blocks may be a "soft" number depending on the operating strategy for the yard, the mix of local versus longer distance blocks, and the total number of railcars that is handled at a yard. As noted earlier, we generally exclude the local blocks from the optimization problem, so that the maximum number of blocks would only reflect the longer distance blocks.

Let $M(\omega)$ = the minimum block size allowed for a block with distance ω.

Let T be the set of traffic. Denote the origin of the traffic $t \in T$ as $o(t) \in Y$ and the destination $d(t) \in Y$. The number of cars associated with a traffic record will be $w(t)$. This notation overlaps the notation for the block origin and destination, but it should be clear from the context when we mean block origin or traffic origin (respectively destination). It is generally assumed that this traffic has been "rolled up" to the serving yards, and excludes the local yards. Further, this formulation is a single commodity formulation, and as a result is generally limited to only carload.

5.8.2.2 Variables

The main variable represents the blocking plan. One way to describe it is as a set of binary variables $\delta_b \in \{0,1\}$ where $b \in \mathrm{B}$. If the block b is included in the block plan then $\delta_b = 1$, otherwise $\delta_b = 0$.

We also have the cost of a classification in yard $y \in Y$ as a non-negative variable P_y. This may appear strange to have the classification cost as a variable. It is natural to consider the very important classification cost to be fixed and known prior to the

start of the optimization. In practice, this is the case and optimization algorithms typically assume the user has good initial values for the classification cost. However there may be circumstances when the classification cost needs to be adjusted during the course of optimization, and hence it will be considered for now as a variable. One example is when ensuring that the capacity of a yard in terms of the number of railcars being handled is respected in an optimal manner.

The block cost is the sum of the classification cost at the origin of the block and the distance of the block, denoted $c(b) = P_{o(b)} + \omega(b)$. P_y is non-negative, $c(b) \geq 0$. As noted earlier, the distance could use weighting factors to reflect routing preferences.

Given the set of active blocks $\hat{\mathrm{B}} = \{b \in \mathrm{B} \mid \delta_b = 1\}$, let the block sequence for a traffic record *t* be based on using algorithmic blocking; it is the shortest path in the graph $(Y, \hat{\mathrm{B}})$ based on the cost *c*(*b*) and denoted as $S\left(t \mid \hat{\mathrm{B}}, P\right) = \left(b_1, b_2, \ldots, b_{k_t}\right)$. In the block sequence, each $b_i \in \mathrm{B}$, and follows the usual rules for a path: $o(t) = o\left(b_1\right), d(t) = d\left(b_{k_t}\right)$, and $d\left(b_j\right) = o\left(b_{j+1}\right)$ for $j = 1, 2, \ldots, k_t - 1$. The notation is meant to explicitly show that the block sequence is dependent on the active blocks and the classification costs, and that the block sequence is an ordered-tuple and not an unordered set.

The cost of a block sequence, $C\left(t \mid \hat{\mathrm{B}}, P\right)$ is the sum of the costs of its components:

$$C\left(t \mid \hat{\mathrm{B}}, P\right) = \begin{cases} \sum_{b \in S\left(t \mid \hat{\mathrm{B}}, P\right)} c(b) & S\left(t \mid \hat{\mathrm{B}}, P\right) \neq \varnothing \\ \infty & S\left(t \mid \hat{\mathrm{B}}, P\right) = \varnothing \end{cases}$$

Note that we do not have a cost for forming a block at a yard, only a cost for using a specific block sequence. Block formation costs are generally treated as zero, and instead we rely on the overall limit on the maximum number of blocks that can be made at each yard. The total cost of the solution is of course dependent on the volume of railcars using each sequence.

5.8.2.3 Constraints

All traffic must be moved:

$$\forall t \in T, S\left(t \mid \hat{\mathrm{B}}, P\right) \neq \varnothing \tag{5.1}$$

Number of blocks originating from a yard must be constrained:

$$\forall y \in Y, \sum_{\left(b \in \mathrm{B} \mid o(b) = y\right)} \delta_b \leq B_y \tag{5.2}$$

To show the number of classifications at a yard, we use the notation that the block sequence for t can be written as $\left(b_1, b_2, \ldots, b_{k_t}\right) = S\left(t \mid \hat{\mathrm{B}}, P\right)$. Using this notation, the constraint for the number of classifications at a yard is:

$$\forall y \in Y, \sum_{t \in \mathrm{T}} \sum_{j=1}^{k_t - 1} w(t) \cdot 1\left\{d\left(b_j\right) = y\right\} \leq C_y \tag{5.3}$$

Every block should have a minimum volume, based on the distance of the block:

$$\forall b \in \mathrm{B}, \sum_{t \in T} \sum_{j=1}^{k_t} w(t) \cdot 1\left\{b_j = b\right\} \geq \delta_b * M\left(\omega(b)\right) \tag{5.4}$$

A number of additional constraints can be introduced, but will not be explored further in this formulation. These include:

- Constraints to support directional activities at a yard, which can be implemented by limiting the set of blocks that can be considered from a specific yard.
- Constraints that require certain blocks to be made, or not made. One can think of this as fixing the integer variables for those specific blocks. An example is the fixing of interchange blocks.
- Constraints on the routing of specific traffic to use specific blocks, essentially fixing part of the path (block sequence) of certain traffic records.

5.8.2.4 Objective

The objective is to minimize total cost over all the traffic records:

$$\min_{\delta_b, P_y} \sum_{t \in T} C\left(t \mid \hat{\mathrm{B}}, P\right) \tag{5.5}$$

As noted earlier, we could introduce weighting factors on the distance costs, and have elected not to include block formation costs. Other formulations have also suggested making use (or non-use) of a yard a factor as well introducing a yard "opening" cost.

Optimization Techniques

There are three levels of techniques used for blocking plan optimization:

1. Automation of the techniques from Sect. 5.6—especially bypass opportunities and low volume block elimination.
2. Additional heuristics.
3. Advanced mathematical programming techniques.

Heuristic Approach

The heuristic approaches find blocking plans by seeking out opportunities to locally improve existing blocking plans by keeping most blocks fixed and only examining a limited number of changes at a time. These approaches rely on several ideas which are explained subsequently:

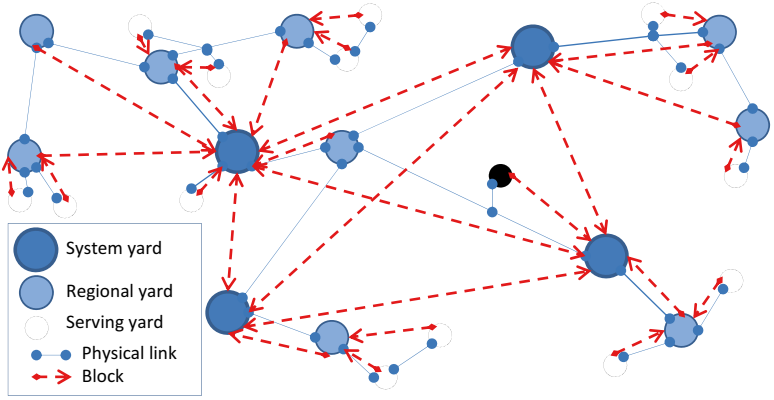
- The ability to create an initial blocking plan.
- Methods for iteratively improving blocking plans.
- The ability to quickly resequence and test out new blocking plans.
- Ability not to get stuck at a local optimum.
- The ability to change the yard penalties if classification capacity constraints cannot be otherwise met.

In many cases they allow for interim solutions that may be infeasible with respect to use of low volume blocks, the number of railcars handled at a specific yard, or the number of blocks formed at a specific yard. These constraints are generally respected in the final solutions, though there can be cases where the requirement that all traffic have a sequence may result in a violation of the low volume block constraint.

Initial Blocking Plan

The most common approach is to build an initial blocking plan based on a hierarchy of yards (or start with the existing plan used by the railroad). Yards can be usually categorized as local, serving, regional or system yards, with the cost per classification decreasing (and the number of classifications per day increasing) for each level of the hierarchy. The concept of the hierarchy is to build a set of bi-directional blocks from each local station to the closest serving yard, from each serving yard to the closest regional or system yard, from each regional yard to the closest system yard, and between all pairs of system yards.

The illustration below is an example where all four system yards have bi-directional blocks between them. Each regional yard has a single block to the closest system yard. Serving yards have a single block to the closest regional yard, with the exception of the serving yard that is in black which has a block to a system yard because that is closer than any regional yard.



One variant of the above is to allow connections from each non-system yard to more than one other yard that is higher in the hierarchy, provided it is within some prescribed distance and you do not need to pass through any other yard that is higher in the hierarchy to reach it. Because of the small number of blocks at other than system yards, this initial solution is usually feasible from a block formation perspective, but there likely will be too many blocks created at some system yards, and sometimes at regional yards as well. Note that the customer locations have been excluded from this process.

Iteratively Improve the Plan

There have been two general strategies for automation of improving an existing plan:

1. Iterative use of bypass opportunities to add potential new blocks, and low-volume analysis to remove blocks (Van Dyke 1986). In this approach, all bypass opportunities are calculated, then one or several of the top opportunities are taken. The bypass opportunities are given a score based on total car-distance, number of classifications, and a cost for violating the two types of yard capacities.
2. Iterative use of rebuilding the blocking plan for a single yard, and iterating through all the yards (Ahuja et al. 2007). This technique is an example of "very large scale neighborhood search." In this approach, for the given yard one block is entered at a time that is deemed the best block based on a score composed of car-distance and classifications. As each block is added, the block sequences are regenerated efficiently.

Both of these approaches rely on algorithmic blocking for determining block sequences. In turn, there is assumed a cost penalty for each classification at a yard, P_y. Often these cost penalties result in too many cars being classified at individual yards and therefore heuristics are used to adjust the cost penalties so that the algorithmic blocking meets the yard capacity.

Resequencing Quickly

Iterative algorithms rely on testing tens of millions—or more—of possible blocking plans. Each test requires evaluating the objective function as described in Eq. (5.5), which involves a full block sequence. Various authors have been reluctant to explain the tricks they developed to resequence quickly, although it is at the heart of the calculations. What is known is that they:

- Use all-pairs shortest path algorithms.
- Are able to restrict the block sequencing to a subset of the traffic at each iteration. One rule is based on logic such as if A–B is a new block, then it will never be used for traffic from B to A so no need to resequence that traffic. More generally, there are many traffic records that should never be classified at a particular yard because doing so would add an unacceptable amount of circuity.

Finding Global Optimum

The iterative techniques discussed above are not proven to be optimal. They stop when no further improvement can be found, but that does not imply optimality—rather it implies that the algorithms are not robust enough to seek better solutions and are stuck in what is known as a local optimum.

There are several methods used to try to move away from local optimum. Two methods, which are often combined, are:

1. Change the constraints (Eqs. 5.1–5.4) into penalties on the objective function. That allows, for example, more blocks than desired to originate from a yard. This may allow iterative algorithms to try out more possibilities then would otherwise be possible.
2. Add some type of randomness into the choice. For example, randomly allow a block into the solution that is economically not very good given all the existing blocks, but later on may prove useful. This is part of the concept of simulated annealing, which has been used in many instances to find better solutions.

There are other techniques that use randomness very successfully in a variety of iterative algorithms that could be applied here. Two popular ones are Tabu Search (Glover and Laguna 1997) and Genetic Algorithms (Simon 2013).

Changing Yard Penalties

Iterative algorithms always have an initial value for the classification cost P_y as described earlier. However, the cost may be too high to allow enough classifications at the yard, or too low, causing the yard to be overwhelmed. The model may not be able to build as many blocks as would be desirable at the yard because the limit on the number of blocks B_y is met well before the limit on the number of classifications C_y is met.

In these cases, the iterative algorithms need to set a trigger that, when over a number of iterations a yard is far away from the limits set, to adjust the penalties. While there is no precise methodology, it is occasionally necessary to make these adjustments to obtain an optimal block design. Typically this means treating the yard capacity constraints as soft (at least for the constraint on the number of railcars), because the violation of these constraints provides important information on how much to adjust the costs or penalties for using the yard.

Advanced Mathematical Programming

Bodin et al. (1980) were the first to produce a mathematical model to create blocking plans, followed by Newton (1996) and Newton et al. (1998). This was followed by Barnhart et al. (2000) that took the work a major step forward to solve blocking

optimization problems of significant size. Their formulation found a near-optimal solution to a somewhat simplified version of the problem. They considered the main constraints—all traffic must obtain a block sequence (Eq. 5.1), the number of blocks (Eq. 5.2) and the number of cars classified at yard are limited (Eq. 5.3). They formulate the problem as a network-design integer program and use advanced mathematical programming techniques including Lagrangian decomposition, column generation, valid inequalities, and dual-ascent to solve the problem.

They start with a large number of potential blocks, and for each traffic flow they find a block sequence within those potential blocks, such that all the block sequences taken together meet the two yard constraints and provide minimal total block sequence cost.

This approach has several issues, however, that need more investigation before it can be used solve real-world blocking problems:

- A necessary constraint for developing a realistic blocking plan is that each block should have a minimum block size (Eq. 5.4). This constraint is not found in their model and while it could be easily placed in their model, it will significantly complicate their Lagrangian decomposition approach.
- The block sequences found may not achieve the routing consistency produced by algorithmic or simple table-based rules because it finds a block sequence for each traffic flow that is governed by capacitation limits on yards. In their case, each traffic flow has a different origin/destination combination. One traffic flow may have a block sequence A–B–C–D–E, another may have a sequence A–B–F–D–G. The inner sequence for the first flow is B–C–D, but it is different (B–F–D) for the second flow because the switching capacity at yard C is met by the first flow, so it needed to alternatively route the second flow through F. Algorithmic blocking in this case will not generally allow two different inner sequences. It is possible to use table-based rules to achieve this outcome, but there will be inconsistencies in the tables—what is the block sequence for traffic from B to D? Is it through C or through F? This may not be a significant issue in practice, but needs to be examined.

  It is possible in their solution to also send half a shipment from B to D via C, and half via F, which also violates using algorithmic or table-based blocking plan designs.
- The authors claim that one part of their approach uses a simplified objective function of only minimizing classifications, and not the total cost of a block sequence. They use this special objective to speed up part of their algorithm. This objective function most likely produces additional circuity.

Despite these issues, we strongly encourage researchers to continue the efforts of using advanced mathematical programming techniques for solving the blocking design problem.

5.9 Additional Considerations

In thinking about the issues related to blocking plan design, optimization, and shipment routing, there are a number of other issues to be considered:

- *Planning Versus Execution Systems*: planning systems and execution systems have different objectives and needs. Real-time systems generally treat the blocking plan as static. The core question they seek to answer is "given that a shipment is at location X, what block should it be assigned to out-bound from X." To answer this question, the system will either use a rule-based look-up process, or an algorithmic routing process. In the planning environment, the goal is more complex. In plan maintenance mode, the systems must support creation, testing, and maintenance of the blocking plan to support the execution systems. To do this, the planners need access to "what if" capabilities, ways to identify possible plan problems and possible plan improvements, tests for plan completeness, and projections of workloads. In addition, planners are likely to periodically take a deeper look at the blocking plan, and seek ways to identify potential broader plan improvements through use of optimization or other improvement techniques.
- *Traditional Problem Separation of Blocks Versus Trains*: At present, most optimization and design strategies separate the blocking plan from the train plan, or approach the problem in an iterative manner. Under this approach, the blocking plan is designed first. The train plan is then created based on the blocking plan. As part of the train design process, issues with the blocking plan may be identified, and used to see if the blocking plan can be improved to yield a better overall solution when the trains are taken into account. This separation is done for two reasons. First, it is dictated in part by the complexity of the problem and the associated difficulties in solving the joint problem. Second, the blocking plan design remains a largely manual process. Even with the use of optimization, the optimizers are only used as a source of ideas or suggestions for plan design, and the final plan usually represents a process of manual review of the optimization results and the selective adoption of the best ideas from the optimization into the final plan. The consequence of this is that wholesale optimization of the blocking plan on a joint basis would be unlikely to produce a result that would be used in the real word, and might be too complex to support manual review. To the extent that joint optimization is possible, this is generally limited to allowing the system to change only a limited number of blocks, both to ensure that the core blocking plan is protected in the optimization process and to make manual review simpler. Such joint optimization strategies are explored in more detail in Chap. 1 on train schedule design.
- *Location-based Routing Control Versus Shipment-based Control*: Under the blocking systems described above, when a railcar is moved, it does not own its own routing plan. Instead, at each location the shipment visits, tables and other systems are examined, and based on the content of these tables, the next location

for the shipment is determined. Thus, the routing plan is "location centric" and not "shipment centric." This had significant advantages in an environment with limited communications, and no fully defined, centralized, computerized operating plan. Each location could have a "blocking book" or "routing guide" and know what to do with each shipment without having to consult with a central authority. Even today, this approach has advantages when shipments are misrouted, or fail to connect to their expected train, because it supports a straight forward way to determine what to do with the shipments. Going forward, it may become more common to instead take a broader network view of the routing process, and then tie the resulting routing to the shipment. When a shipment is processed at a yard, it would then be assigned to a block (or train) not based on local routing instructions, but based on the routing instructions owned by the shipment. A fallback solution will still be required when a shipment falls off its planned routing. This has a number of advantages, including the ability to support reservation type systems, customize routings for individual shipments/customers, and provide a foundation for supporting a dynamic, capacitated routing process.

5.10 Opportunities

Hopefully the reader has gained an understanding of the blocking problem, and the strengths and weaknesses of current approaches to the problem from this chapter, as well as an understanding of where future research and development is needed. While there are many facets to the problem, the authors would like to point out some specific areas for future research below:

(a) Classification table generation problem: as noted extensively in this chapter, most production blocking systems use tables to direct the classification of shipments. Most optimization tools and efficient block sequencing tools use non-table-based algorithms. The reliable translation of these algorithmic routings to table-based solutions that are maintainable and acceptable to railroad planners remains a major challenge that is largely unmet. The authors participated in one such effort that produced a mathematically perfect translation, but was not acceptable to the railroad due to the complexity of the rules that were produced. This complexity resulted in an increase in the total number of rules, made the rules difficult to maintain on a manual basis going forward, were difficult for the planners to understand, and were too different from the historic rules to be acceptable to the planners.
(b) Multi-commodity optimization: most current optimization strategies are single commodity, and cannot take into account the differing needs of each line of business served by the railroad, and the cross-over effects of some traffic operating in dedicated, specialized services and some traffic "falling into" the general carload network. Planning for the movement of grain traffic, which can move in both dedicated trains and in the carload network provides a prime example of this problem.

(c) Joint train/blocking plan problem: trains can be viewed as serving the purpose of moving the blocks in the blocking plan. However, if the blocks cannot be efficiently bundled into trains of reasonable size and complexity, the blocking plan itself can prove to be impracticable. As a result planners typically follow an iterative process where issues in the design of the trains may cause them to make changes to the underlying blocks. While some solutions for train design are capable of suggesting limited changes to the blocking plan, we ultimately would like to see solutions that are of a more integrated nature.
(d) Reservation/capacity management concepts: at present the authors are only aware of one or two railroads on a world-wide basis that use a train level reservation approach to the movement of shipments. Such an approach has the potential to support advanced capacity management concepts that might be able to produce lower cost solutions, improved service reliability, and better overall network management. These concepts are explored in Chap. 4 on car scheduling and simulation.
(e) Local service design: we have repeatedly pointed out that the local service design problem is generally handled manually, and on a separate basis from the more system level blocking plan problem. Tools and techniques for improving the local plan would be very beneficial, particularly given the large percentage of total trip costs associated with local service.

References

Ahuja RK, Jha KC, Liu J (2007) Solving real-life railroad blocking problems. Interfaces 37: 404–419

Barnhart C, Jin HH, Vance P (2000) Railway blocking: a network design application. Oper Res 48(2):1–12

Bodin LD, Golden BL, Schuster AD, Romig W (1980) A model for the blocking of trains. Transport Res 14B:115–120

Crainic TG, Ferland JA, Rousseau JM (1984) A tactical planning model for rail freight transportation. Transport Sci 18:165–184

Glover F, Laguna M (1997) Tabu search. Kluwer Academic Publishers, Norwell, MA

Gorman MF (1995) An application of genetic and Tabu searches to the freight railroad operation plan problem, INFORMS spring meeting

IBM. The optimization of global railways, http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/smarterrail/

Keaton MH (1989) Designing optimal railroad operating plans: Lagrangian relaxation and heuristic approaches. Transport Res 23B:363–374

Keaton MH (1992) Designing railroad operating plans: a dual adjustment method for implementing Lagrangian relaxation. Transport Res 26A:263–279

Kraft ER (1998) A reservations-based railway network operations management system, Ph.D. Dissertation, Department of Systems Engineering, University of Pennsylvania, Philadelphia, PA, UMI Order # 9829930

Kwon OK, Martland CD, Sussman JM, Little PD (1995) Origin-to-destination trip times and reliability of rail freight services in North American railroads, Transportation Research Record, No. 1489, pp 1–8

Little PD, Kwon OH, Martland CD (1992) An assessment of trip times and reliability of boxcar traffic, proceedings of the transportation research forum, 34th annual meeting, vol 1, Arlington, VA

Newton HN (1996) Network design under budget constraints with application to the railroad blocking problem, Ph.D. Dissertation, Auburn University, USA

Newton HN, Barnhart C, Vance P (1998) Constructing railway blocking plans to minimize handling costs. Transport Sci 32(4):330–345

Norfolk Southern Corporation. Next generation car routing system at Norfolk Southern. http://www.informs.org/content/download/239255/2274025/file/SC1.pdf

Railway Age (2014) Obituary for guerdon sterling sines, 1928–2014, railroad computer systems pioneer, railway age, 5 Sept 2014. http://www.railwayage.com/index.php/news/guerdon-sterling-sines-1928-2014-railroad-computer-systems-pioneer.html

Simon D (2013) Evolutionary optimization algorithms. Wiley, Hoboken

Van Dyke CD (1986) The automated blocking model: a practical approach to freight railroad blocking plan development. Transport Res Forum 27:116–122

Van Dyke CD (1988) Dynamic management of railroad blocking plans. Transport Res Forum 29:149–152

Yaghini M, Seyedabadi M, Khoshraftar MM (2012) A population-based algorithm for the railroad blocking problem. J Ind Eng Int, SpringerOpen, 8:8 doi:10.1186/2251-712X-8-8