Summary

With the rise in popularity of rafting, those in charge of the Big Long River want to allow more boats the chance to travel down the river. Here, we present a model to determine an optimal schedule of rafting on the Big Long River that has varying propulsion and trip length options and best uses available campsites. We first analyzed real world data to determine desirable distributions of propulsion types and trip lengths for the schedule. We then developed algorithms to provide schedules that adhere to the distributions while maximizing the number of utilized campsites.

In order to generate schedules algorithmically, we came up with a method using dynamic programming to find legal routes in a schedule. Using this, we were able to iteratively build up schedules one route at a time. In total four algorithms were produced: one naïve algorithm that adds boats to the schedule in a random order, two that add the boats in repeating patterns, and one that uses local search by repeatedly making randomized modifications to the existing schedule and keeping modifications that improved the score of an objective function that takes into account the number of used camps and variety in trip lengths and propulsion types.

We analyzed the algorithms using the objective function score of the schedules they produced. In addition, we developed visualizations of the data to illustrate the output of the algorithms. The algorithms were tested across several cases with varying number of campsites and amounts of noise in the distance between campsites. The local search algorithm for schedule generation proved the most effective, generating a schedule where each campsite was occupied on average 96.1% of the time while maintaining a variety of trip lengths and propulsion types.

Rapid Growth Through Raft Scheduling

Control #15424

February 14, 2012

Abstract

With the rise in popularity of rafting, those in charge of the Big Long River want to allow more boats the chance to travel down the river. Here, we present a model to determine an optimal schedule of rafting on the Big Long River that has varying propulsion and trip length options and best uses available campsites. A visual device is created to assist in analyzing prospective schedules. We also determine how many boats can be added to the rafting season, and advise park managers on ways to find the optimal schedule and river carrying capacity. We create an objective function to be a metric in order to measure algorithm performance and formulate a local search algorithm, applying heuristics to determine a schedule incorporating our stated goals. The results are compared to the output of three other algorithms- one naive and two phasing algorithms.

Our local search method generated a schedule that accommodated 1120 boats and had a variety of propulsion and trip length options. This method is also robust and able to handle changes in number of campsites and inter-campsite distances. Finally, we provide river managers with a memo describing features of our optimal schedule, the best ways to create the best schedules, and the river carrying capacity.

Contents

List of Figures

1 Introduction

White water rafting is enjoyed by thousands of people every year and increasing in popularity. The rivers where rafting takes place can be in remote locations, so camping is required for a significant number of days. At Big Long River, park managers want more people to have this wilderness experience. However, they know too much interaction between rafts or too little variety of options would cause visitor dissatisfaction. To avoid this, they need to know how to accommodate the most trips with a schedule that includes a mix of trip (6 to 18 day) and boat (motor or oar) options which best uses the campsites. The available window for these trips are limited to about 6 months of the year, and a set number of campsites are spread close to uniformly along the 225 mile river.

1.1 Outline of Our Approach

Our aim is to aid the park managers in creating a rafting schedule that maximizes the number of trips while best using the campsites and contains a good variety of trip options. The schedule is intended to be fixed. Customers will reserve a trip for a length in nights, starting at a day in the season, with the propulsion method of their choice. This is in line with practice at commercial rafting outfitters [5]. In this paper, we will

1. Establish assumptions.
2. Develop a naïve algorithm: Random Assignment.
3. Develop ”phasing” algorithms: Basic Phasing and Mirrored Phasing.
4. Develop an intelligent algorithm based on local search.
5. Construct an objective function and heuristics for local search.
6. Examine the behavior of the algorithms’ schedules.
7. Compare the algorithms’ schedules for several case studies.
8. Analyze the intelligent algorithm’s strengths and weaknesses.
9. Give recommendations based on our results.

2 Assumptions

Because there can be very complex variables that come into play, we simplify the problem by making the following assumptions:

Rowboats go on average 4 MPH and motorized boats go on average 8 MPH each day: Although some sections of the river may be faster than others, we assume that over a full day of travel the average speed on that day is the same as the average speed of boats over the whole river.

No two boats can stay at the same campsite: This assumption, given by the problem statement, is a strong constraint on our model; for any particular night, different rafts must stay at different campsites.

Rowboats and motorized boats have a maximum distance/day covered: This assumption stems from the idea that it is undesirable to have rafters travel on the river for very long periods of time in a day. Specifically, we limited boats to traveling for no more than 6 hours of rafting each day. Thus there is a limit on the distance reachable by the boat in a day equal to the speed of the boat time the maximum amount of time a boat should travel each day. Coupled with our assumption about the average speed of boats, this provides a maximum of 24 miles traveled for row boats and 48 miles traveled for motor boats daily.

Campsites are uniformly distributed along the river: This simplifies the model and avoids the danger of infeasible campsite layouts where some sites are so far apart it is impossible for boats to reasonably travel between them in a day. Real-world data on campsite locations along Grand Canyon [6] had close to uniformly distributed site locations, suggesting this is a reasonable assumption. Later, we test our model with small amounts of noise added to the locations of campsites (see Section 6.2: Comparison of the Algorithms)

Boats are always at the scheduled location (start, campsite, or end) at the arranged time: This simplifies the problem because we do not need to take into considerations real-world factors like getting lost, emergencies, cancelled vacations, and is necessary in order for us to draw valid conclusions about boat movements.Then boat routes and schedules can be thought of in terms of campsites alone.

Boats spend no more than 2 nights at the same campsite: For longer trips, rafters might enjoy taking a day off rafting and spending the time relaxing and exploring the nearby area instead. Because of this, our model allows for boats to spend two nights at the same spot. However, we assume boats will not spend a longer time than this in one location.

It is possible for boats to pass each other: Though the problem statement wished for minimal contact between other groups of boats on the river, boats here are allowed to pass each other; otherwise, every boat in the river would be limited to traveling behind the slowest boat.

3 What Makes a Good Model

Based on the problem statement, the agency wants an optimal mix of trips with the intent of allowing more trips down the river. In order to accommodate more trips down the river, either more camps must be built or existing camps must be used more often. Despite this, the agency also wishes for people to enjoy the wilderness experience and avoid interaction with other boat groups as much as possible. Lastly, the agency has made it clear in their request that a varying mix of propulsion type and trip length is a priority. Based on these factors, an optimal schedule has these basic features:

- It maximizes the trips that travel down the river in a season.
- It maximizes the campsite usage.
- It minimizes boat interactions.
- It includes a variety of boat propulsion options and trip length options.

4 Constructing a Model

We will attempt to produce viable schedules by planning rafting trips down the river at varying lengths and propulsion according to several algorithms. The algorithms will work by attempting to build a schedule for boat launch and campsite occupancies across the whole season. A customer reserves a trip from the existing schedule and must then launch at the stated day and camp at the designated sites for each day after that. Various statistics about the schedule are tabulated (such as motorboat to oar boat ratio, campsite usage efficiency, proportions of the various trip lengths). The effectiveness of a given algorithm can then be determined by considering how well it fits our good model criterion.

Using real world data from a popular Grand Canyon rafting company, Arizona River Runners [5], we found approximate expressions for trip length and methods of propulsion offered. The 2012 schedule from April to September showed a motor boat to oar boat ratio of about 4 to 1 and a trip length distribution of $e^{-\frac{(l-6)}{3}}$, where l is the trip length in nights. Since the trip lengths offered did not exactly match the 6 to 18 day trips Big Long River offers, the fitted distribution is only approximate. We assume that the offerings of Arizona River Runners reflect normal values in the industry, so we incorporate these expressions into our algorithms.

According to data of the location of campsites along the Colorado River through the Grand Canyon [6], there are 158 campsites along 295 miles of river. Keeping the same ratio of sites per mile, this corresponds to having 120 campsites along a 225 mile river. However, other rivers listed had a much sparser distribution of campsites. Because of this, we focused on a river with 60 campsites, but also tested out model for rivers with 30 and 120 campsites (see Section 6.2).

5 Algorithm Descriptions

5.1 Definitions

All of the schedule-finding algorithms rely on generating a plan for a trip: a raft propulsion type and trip length in nights and then finding a start night where there exists a route matching the plan and not conflicting with another route. Having a "start night" of n means the raft will first camp n nights after the first night of the season. This leads to the following definitions

Definition 1. *A **trip plan** is a tuple (r, l) where r is the raft type (motor or oar) and l is the number of nights the raft will travel (an integer between 6 and 18).*

Definition 2. *A **route plan** is a tuple (t, s) where t is a trip plan and s is the start night, the first night the raft camps camps.*

We can sort the camps on the river in increasing over of distance from the start location and label them $0, 1, ..., Y-1$ where Y is the number of camps on the river. We denote the location of camp i in miles upriver as x_i. Thus the distance between camps i and j is $d(i, j) = |x_i - x_j|$.

Also recall from our second assumption that each raft type has a maximum distance it can travel each day. We denote the maximum disitnce a raft of type r can travel each day by M_r Using this, we can define a route:

Definition 3. *A **route** is an assignment of camps $c_0, c_2, ..., c_{l-1}$ to a given rout plan $p = (t, s), t = (r, l)$ such that the distance between adjacent camps, $d(c_i, c_{i+1}) \leq M_r$.*

From a route, we can determined the location of the raft in a raft plan satisfying a route plan (t, s) on each night: the raft is at camp c_i on night $s + i$.

Definition 4. *A **schedule** is a set of routes.*

Definition 5. *A **legal route** is a route in a schedule such that every time the raft in the route camps it, does not share the camp with another raft. That is, if the route places the raft at camp c on night n, there must not be another route in the schedule that places a raft at c on night n.*

Definition 6. *A **legal schedule** is a set of legal routes.*

5.2 Finding Legal Routes

This section addresses a natural question that arises from the definitions: **Given a legal schedule S and route plan p, does there exist a legal route satisfying p?**
If such a route exists, the route can be added to the S while keeping it legal. Thus given a list of route plans, we can iteratively build up a legal schedule by adding legal routes to the schedule one at a time.

This problem can be solved using an algorithmic technique called dynamic programming. The algorithm relies on computing the following function for a start night s and raft type t (motor or oar).

$$R_{s,t}(n,c) = \begin{cases} 1 & \text{if a raft of type } t \text{ with start night } s \text{ can reach and spend the} \\ & \text{night at site } c \text{ for night } s+n \\ 0 & \text{otherwise} \end{cases}$$

For all c with $0 \le c \le Y$, n with $0 \le n \le 18$ (the maximum length of a trip). Note that although the camp sites go up to $Y-1$, the function is also computed for $c = Y$, which corresponds to the destination. Given S, we can first compute the function

$$O(n,c) = \begin{cases} 1 & \text{if there is a raft in } S \text{ that spends night } n \text{ at camp } c \\ 0 & \text{otherwise} \end{cases}$$

That is, $O(n,c) = 1$ if campsite c is occupied on night n and 0 if otherwise.

Consider computing $R_{s,t}$ for fixed n and c. A raft can't stay at campsite c if it is already occupied by a different raft.Thus $R_{s,t}(n,c) = 0$ if $O(s+n,c) = 1$. Furthermore, in order for c to be reachable by the raft on night $s+n$, there must be a camp c' before c that is close enough to c so a raft can reach c from c' in one day. This raft must be able to stay there the previous night so we must have $R_{s,t}(n-1,c') = 1$. This lets us explicitly compute $R_{s,t}(n,c)$ for $n \neq 0$ as follows:

$$R_{s,t}(n,c) = \begin{cases} 1 & \text{if } O(s+n,c) = 0 \text{ and there exists a } c' < c \text{ such that } d(c',c) < M_r \\ & \text{and } R_{s,t}(n-1,c') = 1 \\ 0 & \text{otherwise} \end{cases}$$

For the first night, this is slightly different since there is no previous site the raft is coming from. Instead we only need the site to be unoccupied and for it to be reachable from the start. Thus for $n = 0$

$$R_{s,t}(0,c) = \begin{cases} 1 & \text{if } O(s,c) = 0 \text{ and } x_c < M_r \\ 0 & \text{otherwise} \end{cases}$$

Although the definition for R is recursive, it only relies on the value of $R_{s,t}$ for smaller values of n and c. Thus we can compute all values of $R_{s,t}$ by looping going through $n = 0, 1, 2, ..., 18$ and for each n looping through $c = 0, 1, ..., Y$. Once we have R we can generate a route of length l with start night s and raft type t as follows by creating it backwards starting from the destination. This gives the following algorithm.

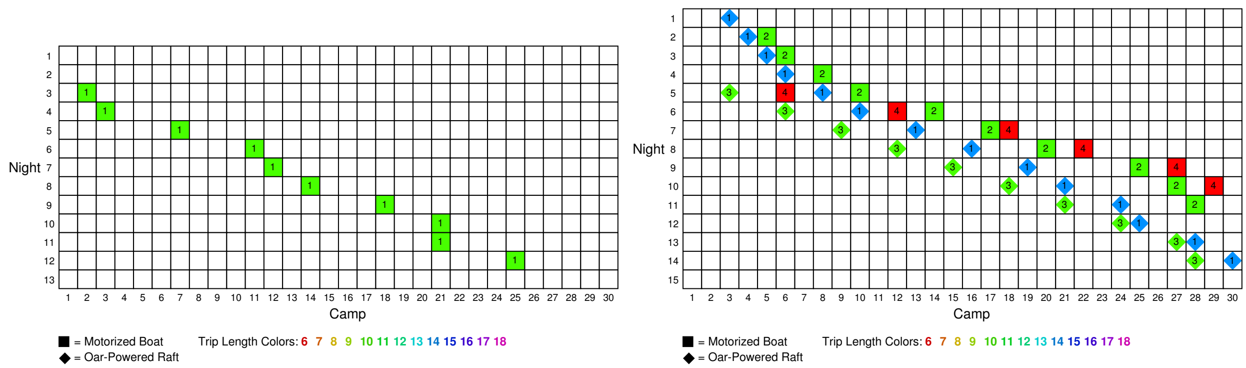
Algorithm 1 FindRoute

Compute $R_{s,t}$ for all c with $0 \le c \le Y$, n with $0 \le n \le 18$ using the recursion described above.
If $R_{s,t}(l,Y) = 0$, **return** NO ROUTE EXISTS.
Let ROUTE be an empty list
Let c = 0
for $n = l-1, ..., 0$ **do**
  Let c' be the largest value $0 \le c' < c$ such that $R_{s,t}(n,c') = 1$
  Add c' to the beginning of ROUTE
  Set $c = c'$
end for
return ROUTE

A fairly straightforward modification of this algorithm allows for the possibility of letting rafts stay up to 2 nights at the same campsite, which was included in out actual implementation.

5.3 Schedule Diagrams

To make analysis of out algorithm's output easier, we created a visual way of representing schedules. The *schedule diagram* is a plot of campsite occupancy on two axes of the day (vertical axis) and the campsite (horizontal axis). The number in the cell indicates the index of the route: so Figure 1 details the entire scheduled route of boat 1 in the season.



(a) Diagram of 1 route. Note a whole day is spent not moving.

(b) Diagram of 4 routes.

Figure 1: Schedule diagrams

Additional features of schedule diagrams become clear as the routing situation grows more complex. Figure 2 is a schedule diagram of four different boats. Since boats 3 and 4 both have their earliest cell at the same day, we can tell they launch on the same day; boat 3 and 1 finish on the same day since their last cell is on the same day. The reduced schedule diagram (as in Figure 3) is used in routing situations that involve large numbers of boats.

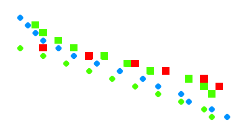


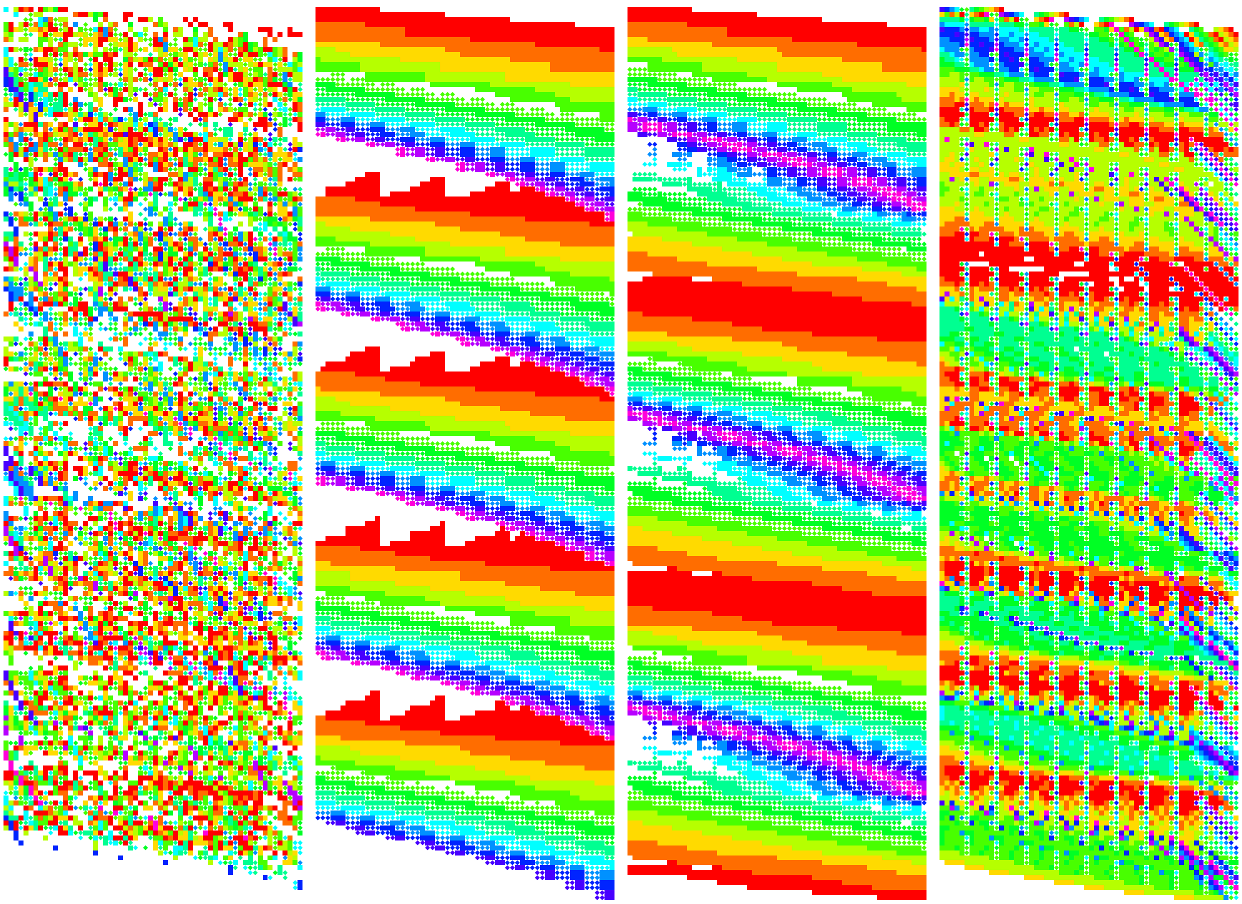
Figure 2: Reduced diagram

The start point and end point of the river are not represented on the schedule diagram, so the schedule diagram only reflects the campsite occupancy along the river.

5.4 Candidate Algorithms

Using the FindRoute procedure, we developed four algorithms to produce schedules - one naïve, two phasing, and one using local search.

1. **Random Assignment:** Assign random boats at the start of every day.
2. **Basic Phasing:** Divide the schedule into phases, and for each phase, assign the shortest routes at the start and routes with longer length later.
3. **Mirrored Phasing:** Divide the schedule into phases, and for each phase, assign the first half to be a phase as in Basic Phasing and the second half to be its reverse.
4. **Local Search:** Use a local search to find the schedule that maximizes our objective function.



(a) Random Assignment (b) Basic Phasing (c) Mirrored Phasing (d) Local Search

Figure 3: Examples of reduced scheduling diagrams

5.4.1 Random Assignment

We begin with the simplest strategy: randomly planning trips at each day.

This naïve strategy serves as a control for the algorithms that follow. The algorithm works by generating a random raft and attempting to add it to the schedule:

Step 1: Generate a raft of random propulsion and route length, with the random variables weighted so that the expectation closely follows that of the real world data described in section 4.

Step 2: Iterate through days in the season, attempting to add the raft to the schedule at the earliest day possible.

Step 3: If the raft cannot be scheduled at any day in the season, the algorithm terminates.

As shown in Figure 3a, this produces a chaotic schedule with a fairly large amount of white space, corresponding to unused camps.

5.4.2 Basic Phasing

The first phasing strategy is a basic attempt to improve upon Random Assignment. The principle behind the algorithm is to note that a schedule which is efficient in terms of camp usage has fewer white cells in its schedule diagram. To that end, we split the season into phases and attempt to schedule rafts in the most gradual order possible within each phase:

Step 1: Generate a list of rafts of fixed size, where propulsion and route length preferences are as close as possible to the real-world data on rafting preferences. Sort the list by route length in either increasing or decreasing order (propulsion type is irrelevant).

Step 2: Iterate through the list one raft at a time from top to bottom: for each raft in the list, iterate through the days to try to schedule the raft at the earliest day possible.

Step 3: When the end of the list is reached, restart iteration at the top of the list. When a raft cannot be scheduled on the last day, the algorithm finishes.

As shown in figure 3b, the schedule does a good job of covering sites except in between phases, in which case there are many unused sites.

5.4.3 Mirrored Phasing

Notice that Basic Phasing has a built in regular period of inefficiency. After each run through the list of rafts, there is a key flaw in restarting the iteration at the top of the raft list. After all, the gradual shift in route length does effectively occupy the campsites near the middle of the season, but the abrupt shift from the longest route length raft to the shortest route length raft when iteration restarts is a cause of major inefficiency. By reducing that loss, Mirrored Phasing is an improvement on Basic. Mirrored Phasing takes the gradual approach strategy and applies it across the entire season, in order to minimize campsite vacancies. The only change from the Basic Phasing algorithm is at the third step: iteration is not restarted at the top of the list, but instead restarts from the bottom and iterates back up to the top. Once it has iterated back to the top, we repeat the whole cycle of top-bottom-top iteration until nothing more can be added to the schedule. As shown in figure 3c, this algorithm does a good job of producing a schedule that covers most of the campsites, although there still is white space in the transition from very slow schedules to faster ones.

5.4.4 Local Search

Local search is an effective technique in answering similar scheduling problems, including the famous traveling salesman problem.[2] The search uses an objective function to quantify how good the schedule is, and looks for an improvement by searching for a neighboring schedule, where neighboring is determined by defined modifications, i.e., heuristics. Since the output of the objective function should show how good the schedule is, we interpret our objectives (maximize boats and variety) as functions and combine them to create a single objective function.

$$\text{We use the objective function } F(s) = C(s)B(s)L(s)$$

$$\text{with the camp score } C(s) = \lambda,$$

$$\text{the boat type score } B(s) = \exp\left(\frac{-\left(\frac{M}{R+M} - T\right)^2}{\alpha_1}\right),$$

$$\text{and the length distribution score } L(s) = \prod_{t=0}^{1}\prod_{l=6}^{18}\exp\left(\frac{-\left(\frac{r_{tl}}{r_t} - T_t(l)\right)^2}{\alpha_2}\right),$$

where we define

$$t = \text{raft type (0 for motor raft, 1 for oar raft)}$$

$$l = \text{length of route in nights}$$

$$r_{tl} = \text{number of routes with raft type } t \text{ and length } l$$

$$r_t = \textstyle\sum_{l=6}^{18} r_{tl} = \text{number of routes with raft type } t$$

$$T_t(l) = \text{target fraction of rafts of type } t \text{ with route length } l$$

$$\lambda = \text{number of filled campsites over whole season}$$

$$\alpha_1,\ \alpha_2 \text{ are constants determining how important the term is.}$$

Note this function increases as the number of occupied campsites increases, and the distributions of boat types and trip lengths approaches the target values, so maximizing this will provide a good score of a schedule. As described in Section 4, we chose the target fraction of motor boats to be $T = 0.8$ and the target fraction of motor boats with trip length l to be given by $T_0(l) = \exp(-(l-6)/3)/\sum_{i=6}^{18}\exp(-(l-6)/3)$ based off real world data. We assumed the desirability of row boat trip lengths would follow the same distribution, but shifted since row boats must travel longer to cover the same distance, giving $T_1(l) = \exp(-(l-10)/3)/\sum_{i=10}^{18}\exp(-(l-10)/3)$. We chose to measure the distance from the current distribution to target distribution in $B(S)$ and $F(S)$ using Gaussian functions ($\exp((x-\mu)/2\sigma^2)$) since it penalizing large deviations at an increasingly higher rate as the deviation increases. We set α_1 and α_2 to be 0.2 and 0.03 respectively after experimenting with various settings.

The local search algorithm starts with an empty schedule and then iteratively tries to improve it. Each iteration, it makes a psuedo-random modification to the old schedule to create a new one. If the new schedule scores higher than the old one according to the objective function, the modifications are kept. Otherwise, the modifications are discarded. After a predetermined number of iterations have passed, the resulting schedule is outputted. The procedure for modifying the schedule is as follows:

1. Chose an interval I of 25 consecutive nights at random from all the nights in the season.

2. Delete all routes in the interval (that is, all routes where the first night of camping is in I).

3. Add n oar-powered raft routes to the interval. The number is dependent on the current ratio of motor boats to oar boats. If the ratio is lower than the target, a smaller number of routes is added, otherwise a larger number is added. The start date of each oar boat is chosen uniformly at random from the interval.

4. For every start night in the interval, keep adding a motor powered raft route with that start night until none exist.

Where the procedure for adding a boat given the boat type and start day is as follows:

1. Compute a preference order of possible length of the trip. Route lengths where the current fraction of routes of that length is far away from the target fraction have higher preference.

2. For all of the lengths in order of decreasing preference, use the FindRoute algorithm to see if adding a route of the given boat type, start night, and length is possible. If it is possible add the route and halt.

We chose to modify blocks of nights at a time because the scheduling of rafts first stopping on a single nights is highly dependent on the scheduling on the other nights near it. This means after a single route is deleted, it is unlikely to be replaceable by a different legal route since there will be a very small legal number of possible legal routes that can be added. Note that the local search is guided by the following heuristics:

- Row boats are added first. This is because row boats are more constrained in how they move (the maximum distance they can move each day is smaller), so adding them is more difficult. Thus they are added first where there are fewer other routes to get in the way. This helps improve $C(S)$.

- The number of row boats added is dependent on the ratio of row boats to oars boats. This helps improve $B(S)$.

- The route lengths of added routes is dependent on the current distribution of route lengths. This helps improve $L(S)$.

A plot of the objective score and its components over the number of iterations the algorithm runs is shown on the next page (Figure 4). All components of the objective function increase as the number of iterations increases, showing that the search strategy was effective at finding states with higher scores. The rate of change in the scores decreases sharply as the number of iterations increases because as the schedule gets better, the chance that a pseudo-random modification will improve it decreases. Since the boat type score and length distribution score functions are not weighted as highly as the Camp Score function (a result of our choice of α_1 and α_2), they tend to fluctuate more as the number of iterations increase, although they also converge to a heigh value.

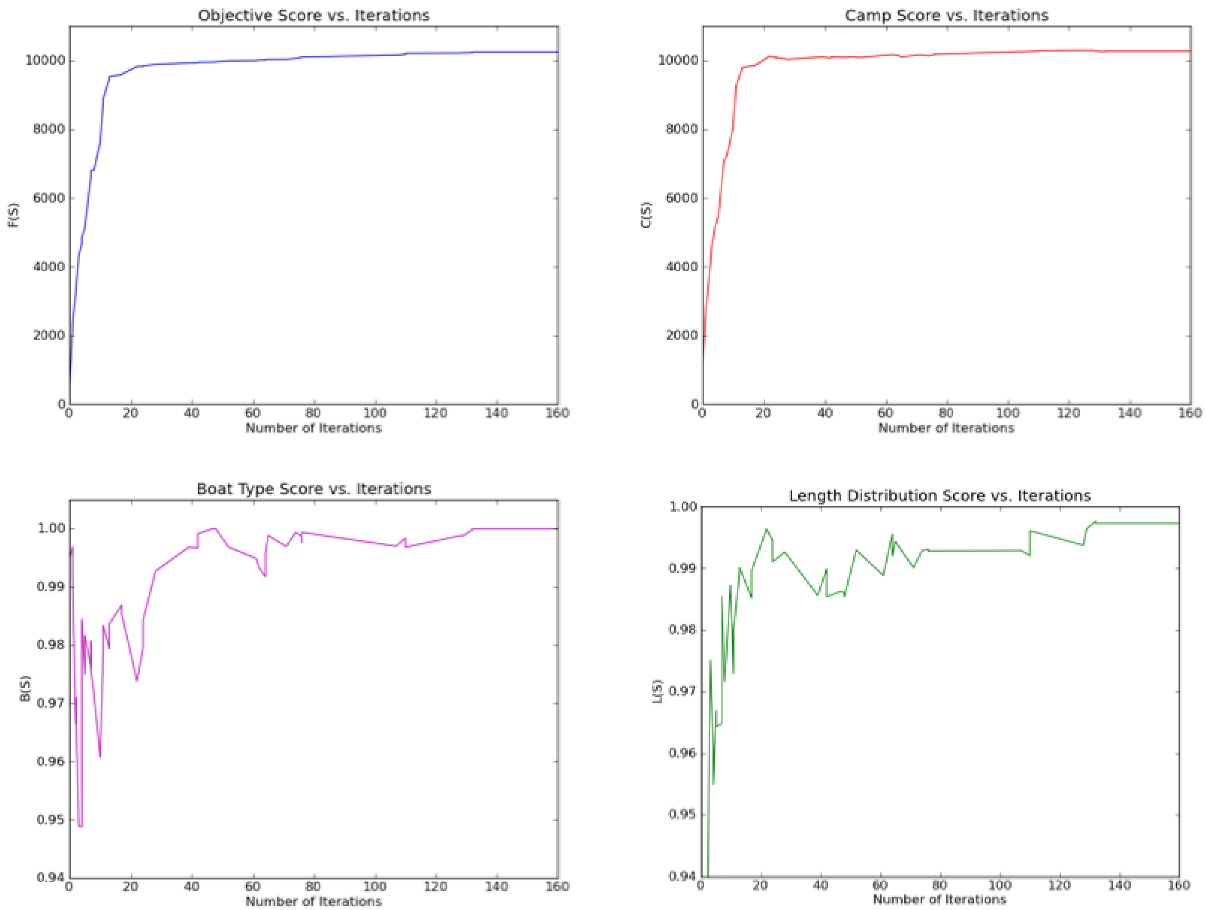


Figure 4: Objective function and component function performance for local searching algorithm

6 Algorithm Results

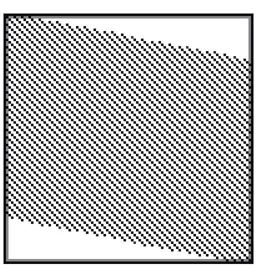


Figure 5: Accessible camps in the season

When finding the capacity, it is useful to calculate the theoretical maximum throughput of the river given campsite occupancy. A perfect maximum capacity schedule would have all camps that can be reached filled everyday of the season. This would occur when the area of the colored region of the schedule diagram is maximized- when the first boat to finish starts at the beginning and has the shorted end day possible, the last boat to finish starts at the last day possible and ends at the last day, and the other boats occupy the space in between. The boat that most fits these first and last boats is the motorized type. So the colored region of the optimal schedule diagram covers $\frac{\#ofcamps\cdot(225miles)}{(8MPH\cdot 6hoursperday)}$ and the total area of a schedule diagram is $(\#ofcamps)\cdot(180days)$: so the maximum boat throughput will have a total efficiency of . The lost 3% represents campsites that either too far from the start of the river to be reached in time for someone to camp there (i.e. a campsite near the end is unoccupied on the second night because boats don?t travel that quickly down the river) or that are too far from the ending of the river for there to be time to finish rafting the river in time for the end of the season (i.e. camping at the first camp possible at the last day of the season is impossible).

With a 225 mile long river and Y campsites, we have prepared the following case studies based on real-world statistics on existing rafting locations[6], adapting the number of campsites to the 225 mile long situation. In all cases, the agency reports to us that there are currently X trips going down the river. Relevant concerns will be establishing a method of scheduling an optimal mix and figuring out how many more trips may be added while staying within our constraints.

6.1 Case Study: A Popular River

Consider the case that the Big Long Riiver is reasonably popular and frequented. A 225 mile river of this type will have 60 campsites fairly uniformly distributed along its length.

6.1.1 Random Assignment Scheduling results

Number of boats			
Boat	#	% of all	Target % of all
Oar	169	0.203	0.200
Motor	665	0.797	0.800
Total	834	1.000	1.000

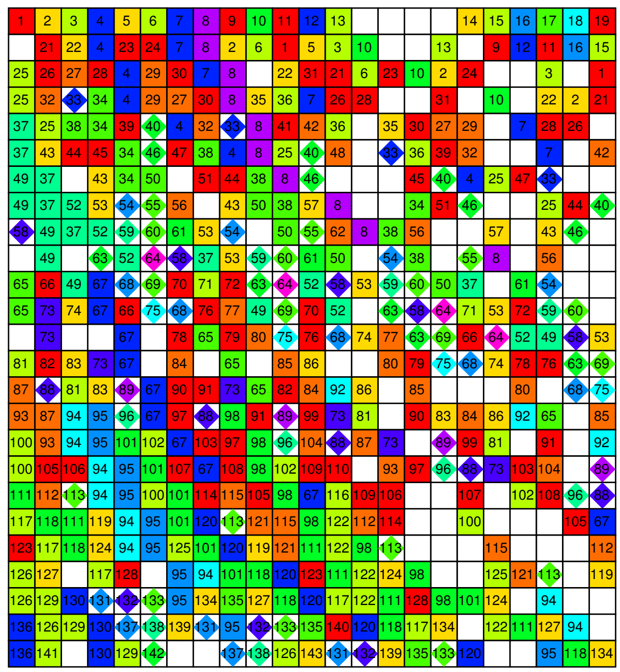


Figure 6: Results of Random Assignment on 60 camps: Boat usage and detail area of example schedule

Objective function data: $F(S) = 7431.7$, $C(S) = 7543$, $B(S) = 1.000$, $L(S) = 0.985$
Campsite usage data: 7543 out of 10800 campsites possible; occupancy over the season: 0.698.

Qualitative Analysis: The Random Assignment algorithm successfully achieves the variety criterion. It has a close match to the desired proportion of boat propulsion in addition to a good match for the duration variety. However, it does not do well at maximizing campsite or river usage. The detail area and the occupancy statistic suggest that the Random Assignment algorithm generates very poor schedules with respect to utilizing campsites effectively and sending more trips down the river. Lastly, the detail area illustrates that the Random Assignment also does not minimize contact with other boats. In our analysis, the Random Assignment algorithm is a good simulation of a system of ”first come, first served” reservations, where patrons lock in a reservation for campsites without any central planning to improve efficiency.

6.1.2 Basic Phasing Scheduling results

Number of boats			
Boat	#	% of all	Target % of all
Oar	202	0.210	0.200
Motor	762	0.790	0.800
Total	964	1.000	1.000

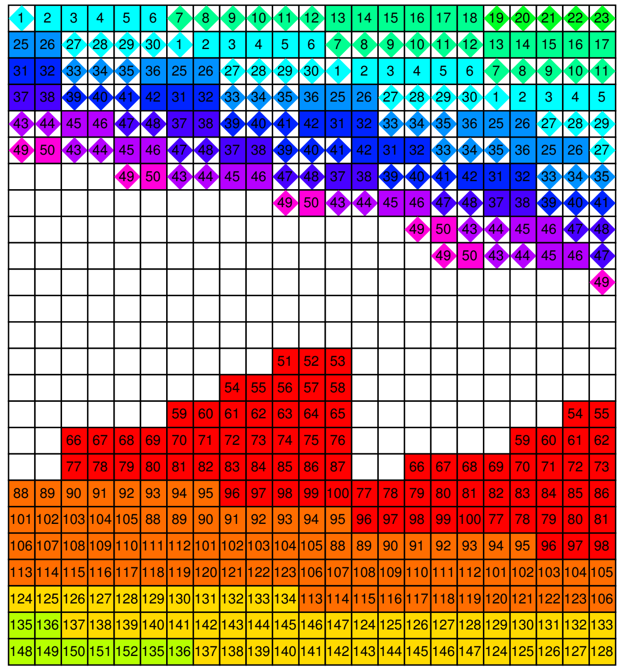


Figure 7: Results of Basic Phasing on 60 camps: Boat usage and detail area of example schedule

Objective function data: $F(S) = 8838.7$, $C(S) = 8951$, $B(S) = 0.997$, $L(S) = 0.990$
Campsite usage data: 8951 out of 10800 campsites possible; occupancy over the season: 0.829

Qualitative Analysis: The schedule generated from Basic Phasing performs worse when it comes to variety, in both propulsion and route length, but does a much better job of achieving increased campsite and river usage. The detail area illustrates the unused campsites: they occur exactly in the time window when the long trips end and the short trips begin again. Lastly, when it comes to minimizing boat interaction Basic Phasing does well. For example, look at the boats from range 124 to 133 in the detail area. They maintain their order between successive days, a marked improvement over the chaos of Random Assignment.

6.1.3 Mirrored Phasing Scheduling results

Number of boats			
Boat	#	% of all	Target % of all
Oar	222	0.207	0.200
Motor	849	0.793	0.800
Total	1071	1.000	1.000

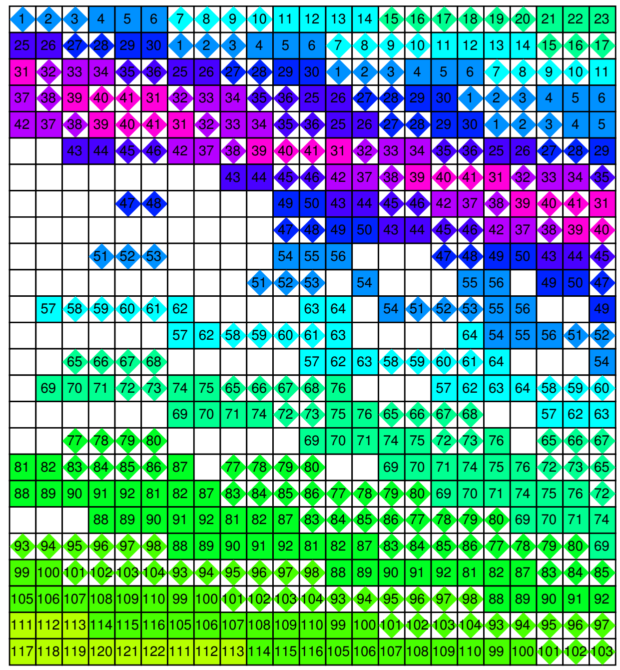


Figure 8: Results of Mirrored Phasing on 60 camps: Boat usage and detail area of example schedule

Objective function data: $F(S) = 9859.2$, $C(S) = 9960$, $B(S) = 0.998$, $L(S) = 0.992$
Campsite usage data: 9960 out of 10800 campsites possible; occupancy over the season: 0.922.

Qualitative Analysis: Mirrored Phasing achieves better variety over propulsion than Basic Phasing. It also schedules additional long routes than the other two, algorithms. The detail area shows how the gap between long routes no longer exists as it did in Basic Phasing: there are unused campsites, but it is a significantly better schedule with respect to maximizing campsite and river usage. Lastly, it offers the same advantages as Basic Phasing when it comes to minimizing boat interaction. The same orderly structure from Basic Phasing is observed.

6.1.4 Local Search Scheduling results

Number of boats			
Boat	#	% of all	Target % of all
Oar	217	0.194	0.200
Motor	903	0.806	0.800
Total	1120	1.000	1.000

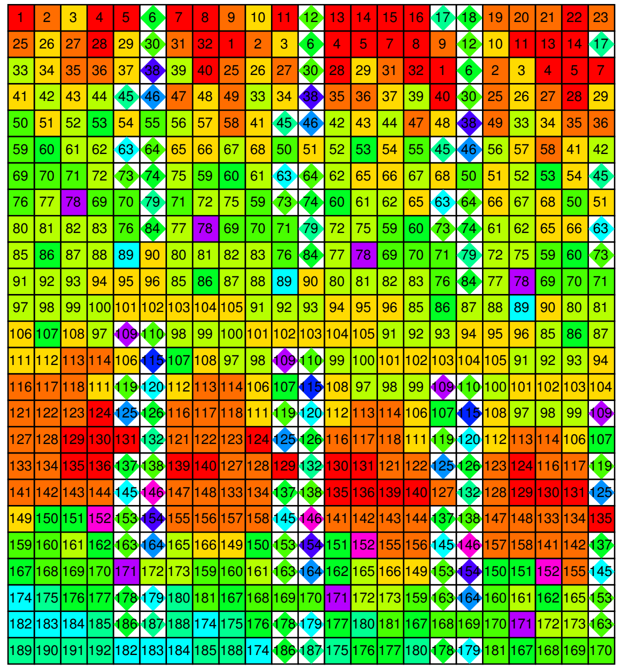


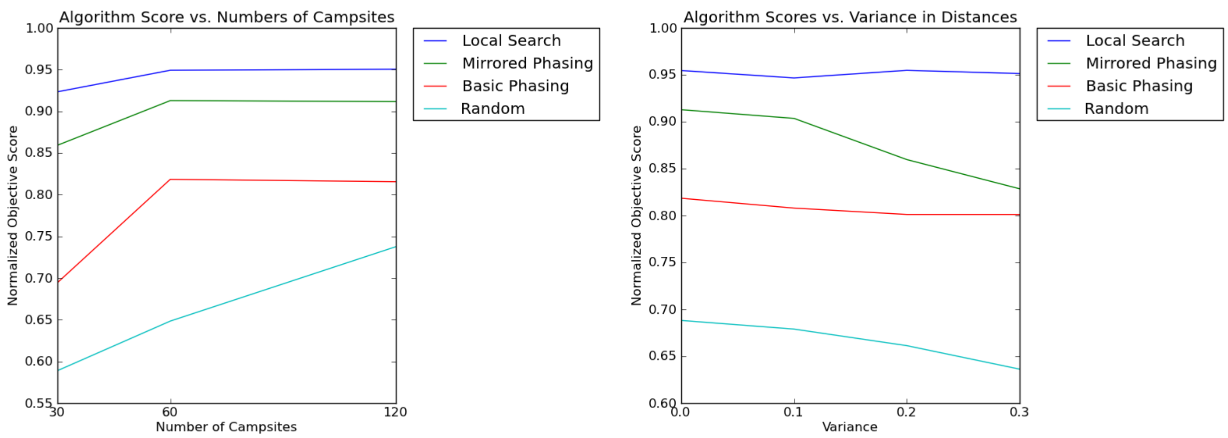
Figure 9: Results of Local Search on 60 camps: Boat usage and detail area of example schedule

Objective function data: $F(S) = 10309.1$, $C(S) = 10375$, $B(S) = 0.999$, $L(S) = 0.995$
Campsite usage data: 10375 out of 10800 campsite occupancy over the season (0.961).

Qualitative Analysis: Right away in the detail area we see how local search has found a schedule with very little inefficiency. Refer to the reduced schedule diagrams in section 5.2: Candidate Diagrams; with local search campsite usage (and therefore possible additional trips down the river) is very high. For variation in propulsions and route lengths, local search outperforms both Mirrored and Basic Phasing, although it does not match Random Assignment in that regards. Local search, however, does not attempt to minimize boat interaction at all.

6.2 Comparison of Algorithms

These algorithm results can be compared, with the number of campsites and variance of distance between campsites varied. The normalized objective score (algorithm score) is the schedule's objective score over the theoretical maximum objective score.



(a) Campsites (b) Distances

Figure 10: Algorithm Comparisons

Because the number of campsites, Y, in the problem statement is not defined, we can investigate how the algorithms perform for various Y values. Testing for $Y = 30$ (few camps), $Y = 60$ (many camps), $Y = 120$ (very many camps), it is clear the relative performance of all the algorithms with respect to each other is consistent throughout, and our local search algorithm outperforms the other three in all cases. The difference between local search and its nearest competitor, mirrored search, is around 4% to 6%. The shapes of the all but the random algorithm are similar in that $Y = 60$ shows an improvement on $Y = 30$, but $Y = 60$ and $Y = 120$ are about equal. While these algorithms seem to have plateaued in performance by $Y = 120$, it appears the random algorithm performance is still increasing. Still, local search seems to be the most effective and flexible for dealing with any number of campsites; whether campsites are added or removed, local search makes a nearly optimal schedule.

When Gaussian noise was added to the distance between camps, making the campsite distribution close to but not exactly uniform, our local search algorithm once again outperforms the others. The relative performance of the algorithms actually matches the findings above, but performance for the random and mirrored phasing decreases for increasing variance, while local search and basic search have more steady performance. This suggests our local search algorithm is more effective for realistic conditions such as variance in distances between campsites.

7 Conclusion

7.1 Summary of Results

Random Assignment: Random Assignment is inefficient because it lacks foresight into future routes. This algorithm is not recommended and is presented as a control for the next three algorithms.

Basic Phasing: Basic Phasing serves as an improvement on Random Assignment, but our analysis above has shown that it is outperformed by Mirrored Phasing based on our objectives.

Mirrored Phasing: Mirrored Phasing is a very successful scheduling algorithm by our measure and is only outperformed by Local Search in terms of campsite usage. An additional benefit of Mirrored Phasing is that the schedules it generates are simpler in that they could be created by hand and without access to computational resources.

Local Search: Local Search offers the highest available efficiency on campsite usage, but requires computational resources and may not minimize boat interactions.

Local Search is the recommended algorithm because it is robust and achieves increased trips that utilize campsite resources in the best way possible. Mirrored Phasing is a feasible option for rafting companies that prioritize interaction between boats over meeting increased demand.

7.2 Strengths and Weaknesses

The model has the following strengths and weaknesses:

Strengths

1. The percent of used campsites over the whole season is near the theoretical maximum.
2. There is a good mix of motorized and oar boats and of trip lengths between 6 and 18; the distributions closely follow those found in real rafting companies.
3. These results hold as Y, the number of campsites on the river changes.
4. These results hold when the distance between adjacent campsites has a fairly high degree of variance.
5. The algorithm used to generate schedules is very generalizable. The desired distribution of row boats vs motor boats, desired distribution of trip lengths, and importance of matching those distributions can be changed and the algorithm will produce a schedule matching those criteria.

Weaknesses

1. The schedule is complicated and does not follow a well defined pattern, which may make it difficult to administer and confusing for people trying to choose a route. In contrast, the basic phasing and mirrored phasing algorithms produce a more well organized schedule.
2. Boats pass each other with fairly high frequency. In contrast, the schedules produce by the basic phasing and mirrored phasing algorithms have almost no boat crossings since most boats on the river at the same time have the same speed

7.3 Improving the Model

While the algorithmic approach offers robustness when it comes to campsite number and river distance, it is built upon some nontrivial constraints that offer room for improvement:

Adding in boat interaction to the objective function: At present our analysis of boat interaction relies on the visual presentation of the schedule diagram. A scoring method for boat interaction would allow comparison between algorithms on a more quantitative level.

Allowing a group to camp for more than 2 nights in a row: Allowing for groups to stay in the same place for many days would allow other boats to pass groups without adding to boat interaction.

Allowing multiple groups to share the same camp: While it is undesirable in the interests of preserving the experience of Big Long River, if demand for rafting trips grows then it may be necessary to allow, but discourage, the practice of sharing a camp. Our model can be extended by modifying the objective function so that campsites can be shared but at a very large penalty.

8 Appendix: Data Tables

Route length	Oars scheduled	Oars desired	Motors scheduled	Motors desired
6	N/A	N/A	188 (0.283)	0.287
7	N/A	N/A	139 (0.209)	0.206
8	N/A	N/A	110 (0.165)	0.147
9	N/A	N/A	71 (0.107)	0.106
10	52 (0.308)	0.298	48 (0.072)	0.076
11	34 (0.201)	0.214	27 (0.041)	0.054
12	23 (0.136)	0.153	28 (0.042)	0.039
13	25 (0.148)	0.110	21 (0.032)	0.028
14	13 (0.077)	0.079	11 (0.017)	0.020
15	9 (0.053)	0.056	7 (0.011)	0.014
16	6 (0.036)	0.040	9 (0.014)	0.010
17	6 (0.036)	0.029	2 (0.003)	0.007
18	1 (0.006)	0.021	4 (0.006)	0.005

Table 1: Random Assignment Route Length Distribution

Route length	Oars scheduled	Oars desired	Motors scheduled	Motors desired
6	N/A	N/A	210 (0.276)	0.287
7	N/A	N/A	155 (0.203)	0.206
8	N/A	N/A	110 (0.144)	0.147
9	N/A	N/A	80 (0.105)	0.106
10	55 (0.272)	0.298	60 (0.079)	0.076
11	40 (0.198)	0.214	40 (0.052)	0.054
12	30 (0.149)	0.153	30 (0.039)	0.039
13	25 (0.124)	0.110	25 (0.033)	0.028
14	15 (0.074)	0.079	15 (0.020)	0.020
15	15 (0.074)	0.056	15 (0.020)	0.014
16	10 (0.050)	0.040	10 (0.013)	0.010
17	8 (0.040)	0.029	8 (0.010)	0.007
18	4 (0.020)	0.021	4 (0.005)	0.005

Table 2: Basic Phasing Route Length Distribution

Route length	Oars scheduled	Oars desired	Motors scheduled	Motors desired
6	N/A	N/A	225 (0.265)	0.287
7	N/A	N/A	174 (.0.205)	0.206
8	N/A	N/A	126 (0.148)	0.147
9	N/A	N/A	90 (0.106)	0.106
10	60 (0.270)	0.298	66 (0.078)	0.076
11	48 (0.216)	0.214	48 (0.057)	0.054
12	36 (0.162)	0.153	36 (0.042)	0.039
13	24 (0.108)	0.110	24 (0.028)	0.028
14	18 (0.081)	0.079	18 (0.021)	0.020
15	12 (0.054)	0.056	12 (0.014)	0.014
16	12 (0.054)	0.040	12 (0.014)	0.010
17	6 (0.027)	0.029	12 (0.014)	0.007
18	6 (0.027)	0.021	6 (0.007)	0.005

Table 3: Mirrored Phasing Route Length Distribution

Route length	Oars scheduled	Oars desired	Motors scheduled	Motors desired
6	N/A	N/A	238 (0.264)	0.287
7	N/A	N/A	180 (0.199)	0.206
8	N/A	N/A	131 (0.145)	0.147
9	N/A	N/A	93 (0.103)	0.106
10	66 (0.304)	0.298	80 (0.089)	0.076
11	46 (0.212)	0.214	50 (0.055)	0.054
12	33 (0.152)	0.153	34 (0.038)	0.039
13	23 (0.106)	0.110	29 (0.032)	0.028
14	17 (0.078)	0.079	18 (0.020)	0.020
15	12 (0.055)	0.056	18 (0.020)	0.014
16	9 (0.041)	0.040	19 (0.021)	0.010
17	6 (0.028)	0.029	8 (0.009)	0.007
18	5 (0.023)	0.021	5 (0.006)	0.005

Table 4: Local Search Route Length Distribution

References

[1] Conway, R.; Maxwell, W.; Miller, L. *Theory of Scheduling*: Addison-Wesley Publishing Company. 1967.

[2] Pinedo, M. *Scheduling: Theory, Algorithms, and Systems.* 3rd Edition. Springer. New York, NY. 2007.

[3] Baker, K.; Trietsch, D. *Principles of Sequencing and Scheduling.* John Wiley & Sons. Hoboken, NJ. 2009.

[4] Chretienne, P.; Coffman, E.; Lenstra, J.; Liu, Z. *Scheduling Theory and its Applications.* John Wiley & Sons. Chichester, West Sussex. 1995.

[5] Raft Arizona. 2012. Arizona River Runners. Feb. 10, 2012. <http://www.raftarizona.com/rates-dates/default.asp>.

[6] White Water Campsites. Hansen, Will. 2009. White Water Campsites. Feb. 12, 2012. <http://www.whitewatercampsites.com/>.

9 Memo

TO: Big Long River agency
DATE: February 13, 2012
SUBJECT: Scheduling an Optimal Mix of Trips: Results

This memo addresses how to determine the carrying capacity of Big Long River, ways to develop the best schedules, and recommendations on how to utilize campsite resources in the best way possible. The best schedule is one that has a mix of trips down the river of varying duration and propulsion type and with minimal contact with other boats on the river.

Increased demand for rafting means Big Long River, like other rafting destinations, must improve the efficiency of their rafting trips in order to service demand in the 6 month long rafting season. Big Long River already has a system of established campsites at regular intervals down the river.

To determine the carrying capacity of Big Long River, a metric of campsite occupancy must be defined: $(\#camps \cdot seasonlength)$. However, this includes camps that are impossible to schedule raft trips to; either the season ends too quickly for a boat to make the trip from that camp to the end of the river, or the camp is too far from the start of the river to be reached in a short amount of time. This leads to the percentage $\frac{\#ofcamps \cdot (225miles)}{(8MPH \cdot 6hoursperday)} \cdot \frac{1}{(\#ofcamps) \cdot (180days)} \cdot 100\%$: $\approx 97\%$.

To determine the best method of scheduling, several algorithms were constructed and tested against each other using a variety of constraints: campsite usage, trip variety, and raft interaction. After analyzing the data, we have determined that a local search algorithm will best develop a schedule that fits the needs of the Big Long River agency.

The main scheduling algorithms that would be of benefit to Big Long River in this study are Mirrored Phasing and Local Search. Mirrored Phasing is a scheduling strategy that works by making sure all the boats scheduled in a specific time frame are close together in terms of how long their trip is. Local Search works by applying a local search algorithm to an objective function that is set to maximize campsite usage, trip variety, and raft type variety.

Local search is recommended due to its performance in scheduling. For campsites over the entire season, local search produces 96.1% occupancy: very close to the theoretical maximum. In addition, the schedule generated by local search provides a good variety between trip lengths and boat types: our results with local search are very close to what market research suggests are real-world preferences for trip lengths and propulsion types. The downside of local search is that it will have a higher degree of interaction between boats. If the current situation at Big Long River is such that demand vastly outstrips current supply of schedules, then local search is the best choice. However, if the need for increased trips down the river is outweighed by the desire to maintain a scenic ride and a natural experience, then Mirrored Phasing scheduling is recommended.

Our results are presented for your consideration. The local search scheduling algorithm does the best job of scheduling an optimal mix of trips at a very high level of efficiency in campsite usage. The mirrored phasing scheduling algorithm offers improved campsite usage without comprimising on interaction between boats.