Control #15421

SUMMARY

In this paper we present a method of optimally scheduling trips travelling down Big Long River. Our model takes the number of campsites and the relative distribution of types of trips as parameters and generates a schedule that maximizes the number of trips and determines when each should be launched. We determine the optimal launch date of each trip by modeling the collection of trips as a chemical system of crystallizing molecules and use the method of Simulated Annealing to generate a launch schedule in which human interactions are minimized. Starting with a mathematically derived upper bound on the number of trips we reduce the number of trips until we find an amount which allows the model to produce a valid schedule. In this paper we take into account user comfort and emergency needs by scheduling three campsites per trip per night. This paper also discusses the flexibility of our model and presents sample output for a user-defined input.

Crystallization of Metal Particles as a Model for Optimal River Rafting

Control #15421

February 13, 2012

In this paper we present a method of optimally scheduling trips travelling down Big Long River. Our model takes the number of campsites and the relative distribution of types of trips as parameters and generates a schedule that maximizes the number of trips and determines when each should be launched. We determine the optimal launch date of each trip by modeling the collection of trips as a chemical system of crystallizing molecules and use the method of Simulated Annealing to generate a launch schedule in which human interactions are minimized. Starting with a mathematically derived upper bound on the number of trips we reduce the number of trips until we find an amount which allows the model to produce a valid schedule. In this paper we take into account user comfort and emergency needs by scheduling three campsites per trip per night. This paper also discusses the flexibility of our model and presents sample output for a user-defined input.

Contents

1. Assumptions and Representation

a. Global Assumptions

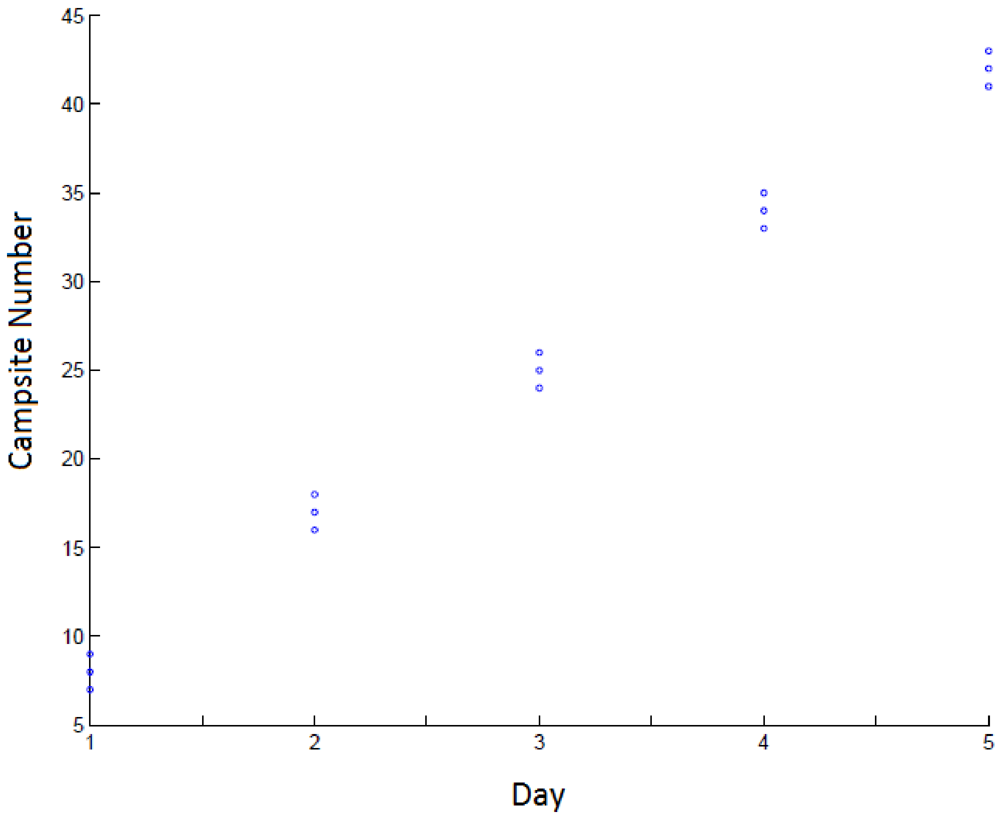
As per the problem statement we assume that all campsites are evenly spaced, with the distance between two consecutive campsites given by $d = (225\, miles)/Y$ where Y represents the total amount of campsites. We can then represent the schedule as a grid with *x*-axis representing time in days and *y*-axis representing distance (measured in multiples of *d*). Therefore, each point *(x,y)* with integer coordinates represents campsite number *y* on day *x*, where campsite 1 is closest to First Launch and campsite *Y* is closest to Final Exit.

Sample values used to generate data are described in Section 4a: "Sample Input"

b. Modeling a Single Trip

The ultimate goal of the model is to increase the total number of trips while preserving quality and variety of trips. When deciding how to best model a trip we came to a compromise between these two by enforcing that each trip has a specific, pre-determined total length that is determined by the camp director before the start of the season. In our model we have decided to set aside three campsites per active day of each trip. The first "base" campsite is determined by the average pace of the trip. Given the length of the trip we can determine the average amount of miles the trip must travel per day, and round down to the nearest campsite to determine at which campsite they will spend the night. To give campers variety and choice in pace, our model also guarantees that the campsites immediately before and after the "base" campsite are also unoccupied. This allows boaters to choose between three potential campsites every day, meaning that they have three paces to choose between every day. Setting aside three campsites for each day of a trip also guarantees that never are more than a third of the campsites filled, which reduces the amount of people on river and allowing tourists a closer connection to nature and less interaction with other trips.

As we discuss our model notice that we completely disregard the fact that there are two different kinds of boats. The length of a trip entirely determines the trip's average speed and therefore which campsites must be set aside for it. However, as will be discussed in Section 4A, "Results Generated By The Current Model", the kind of boat chosen (motorboat or oar-powered) will affect the length of the trip, and is therefore accounted for in our model.



The above represents a trip of length 6 (the trip will spend 5 nights in campsites) that starts on day 1 and ends on day 5. Each point represents a campsite that is set aside for this particular trip on the day specified by the x-axis.

2. Modeling a System of Trips

a. Annealing and Crystallization in Chemistry and Metallurgy

Crystallization of a material plays a critical role in Chemistry and Metallurgy and is the procedure of converting a material from a 'impure form' to a 'crystalline form'. Crystallization arises when a material in 'impure form' is heated to a very high temperature, allowing individual molecules to transition between almost any configurations. As the material is cooled, the molecules begin to favor lower-energy configurations, and thus begin to collect in local and global minima. However, since the temperature is lowered *slowly*, molecules generally do not collect in local minima because a high temperature allows them to jump out of a local minimum with much more ease than out of a global minimum. Thus, as the energy is lowered, the molecules of the material collect in the global minimum, yielding the crystalline form of the material. This is how crystals are made from crude materials in Chemistry and Metallurgy.

b. Energy

Given a system of trips (read "collection of trips") with determined lengths, we can calculate the coordinates of each of the campsites that will be set aside for each trip. Define the **energy** of the system as the amount of times that two trips require the same campsite on the same day. For example, if our system consists of two trips that are perfectly overlapping (they start on the same day and last the same amount of time) then each campsite used by one trip will also be used by the second trip. Thus the energy would be twice the total number of campsites required by one trip.

c. Simulated Annealing: Overview

We treat a system of trips similarly to the materials described above. A random system is generated where N trips are distributed randomly among D total days. The energy of the system is calculated, and we define a 'neighbor' of this system as a system that can be derived by changing the start date (therefore moving all campsites) of *one* trip by no more than five days. Thus, each system has a total of ten times the number of trips neighbors. At each iteration of the process, the current system picks a random neighbor, and if the energy of the neighbor is lower than the energy of the current state, the system moves to this new state. Otherwise, if the energy change is unfavorable (ie the energy of the neighboring state is higher than the energy of the current state) then the probability of moving to this high-energy state is determined by a function of the Temperature of the system (details of the form of the probability function can be found in Section 2f).

This process, aptly named "Simulated Annealing", ensures that the system is always make a net move towards a 0-energy system (where there is no overlap between trips) but also does not get stuck at local minima. If the system moves to a relatively low-energy neighbor) then there is a probability (as a function of temperature) that the system will move out of the local minimum. We chose to use the Simulated Annealing optimization procedure over a greedy-algorithm because of the inherently large amount of possible configurations of Trips and therefore large amount of local minima.

d. Input Parameters:

- *Y*: Number of campsites. This is determined by the camp director.
- Proportion of each kind of trip: this is determined by the camp director based on demand for each kind of trip. If the only goal is to maximize the total number of trips (without regard to providing a variety of trips) then the director should choose to launch exclusively six day trips (because shorter trips occupy less campsites overall, and therefore allow for more total trips).
- Length of boating season: For simplification, we assume six months is exactly 180 days, but that might not fit exactly with reality. Still, the process we use to find the optimum schedule works for any number of days.

e. Simulated Annealing: Specifics

According to [2], a simulated annealing algorithm has the following elements:

1. A finite set S representing all of the possible solutions. In our case, this is the set of all possible arrangements of start dates of all of the trips, regardless of whether the arrangement adheres to the restriction that no two trips can use one campsite at the same time.
2. A function $J{:}S \rightarrow R$ which represents the energy of each member of S. Our implementation of the energy function is described in Section 2b.
3. For each element *i* in S, a subset of S called S(*i*) which defines the "neighbors" of *i*. In our case, the neighbors of *i* are all of the elements of S where the start date of exactly one trip is changed by no more than 5 days.
4. A cooling schedule function defined as $T{:}\ N \rightarrow 0$ which defines the probability that we accept a move from our current state to a worse state.
5. An initial state. In many Simulated Annealing implementations this initial state is randomly selected, but in our implementation we create an initial state where trips are distributed evenly throughout the possible starting days.

To improve runtime their implementation also includes a matrix with values representing the probability of choosing each of the neighbors of the current state. Due to time constraints we omitted this modification and chose which neighbor to move towards randomly. However, this should not change whether the process converges, only runtime.

f. The Probability Function

As described by several papers, simulated annealing functions by iterative improvements of the initial state [1]. At each iteration, the algorithm randomly chooses an element from the neighbors of the current state; if the chosen neighbor has a lower energy the algorithm iterates again with initial state equal to the neighbor state. If the chosen neighbor state has a higher cost, then the algorithm decides whether to accept it as the new current state with probability

$$P = \exp(\frac{\Delta E}{T(I)})$$

where ΔE is the difference between the energy of the neighbor state and the current state (this is negative when the neighbor state has higher energy than the current state) and *T(I)* is the "temperature" of the system as a function of current number of iterations the algorithm has gone through. The purpose of the energy function is to prevent the algorithm from getting stuck at local minima while also being less likely to accept worse solutions as time goes on, thus pushing the system to the global minimum of energy 0.

g. The Temperature Function

The probability function is strongly dependent on the choice of the temperature function *T*. According to Theorem 1 from Bertsimas and Tsitsiklis [2], simulated annealing using a temperature function of the form T(t) = d/log(t) is guaranteed to converge for d > d*, where d* is the maximum "depth" of the local minima of J. Thus we have chosen the temperature function

$$T(t) = \frac{30}{\log(t)}$$

The number 30 was determined empirically by testing many values for *d*. We also tested temperature functions the decreased more quickly, such as the square root function, combinations of power series, and several combinations of these functions. In most cases the resulting temperature function declined to quickly, allowing the system to get stuck at local minima.

Our usage of simulated annealing is not entirely typical. The intended purpose of the Simulated Annealing algorithm is to determine the state with minimum energy when the absolute minimum energy attainable *is not known.* This means that the temperature function must grow very slowly to guarantee that the system does not get stuck at a local minimum. In our case we know that the absolute minimum energy is 0, and so we can stop the simulation as soon as we find a system whose energy is 0. This vastly reduces computation time, which is one of the biggest issues with the algorithm

3. Creating a Schedule

a. Upper Bound

Consider the lower $\frac{1}{6}$ of the grid. After the first day, our model requires that a trip travel at least $\frac{Y}{18} - 1$ campsites (this is the lower bound for an 18-day trip) and at most $\frac{Y}{6} + 1$ campsites. Thus, every trip must stop for the night somewhere in the box determined by $[1, D] x \left[\frac{Y}{18} - 1\ ,\ \frac{Y}{6} + 1\right]$, where *D* is the total days in which trips can run(6 months is approximately D = 180). The total amount of campsites in this area is given by:

$$A = D\left(\frac{Y}{6} + 1 - \frac{Y}{18} + 1\right) = D\left(\frac{Y}{9} + 2\right)$$

Notice that it is impossible for any trip to start within 6 days of the last day of the season, so the triangle of base 6 and height $\frac{Y}{9} + 2$ is inaccessible to trips. Thus we are left with a total of

$$A - 3\left(\frac{Y}{9} + 2\right) = (D - 3)\left(\frac{Y}{9} + 2\right)$$

possible first-stop campsites.

Now consider a trip *T* with total length *d*. Of the campsites in the area described above we will need to set aside at least $3\ (floor\left(\frac{d}{6}\right))$ campsites for use by a trip (i.e. a trip of length 6 will need three campsites and a trip of length 18 will need 9). Using this, we can calculate an upper bound on the number of trips that we can launch. Let P represent the total number of trips and p_d represent the number of trips of length *d*. We can write

$$\sum_{d=6}^{18} p_d = P$$

In our model the user is allowed to input the desired frequency of each trip. This means that each value $c\_d \equiv\ p\_d/P$ is given as input. Thus we have that the total number of campsites used (of the ones in the area described above) is given by:

$$\sum_{d=6}^{18} 3 \cdot p_d \left\lfloor \frac{d}{6} \right\rfloor \leq (D-3)\left(\frac{Y}{9}+2\right)$$

$$3P \sum_{d=6}^{18} \frac{p_d}{P} \left\lfloor \frac{d}{6} \right\rfloor \leq (D-3)\left(\frac{Y}{9}+2\right)$$

$$P \leq \frac{(D-3)\left(\frac{Y}{9}+2\right)}{3\sum_{d=6}^{18} c_d \left\lfloor \frac{d}{6} \right\rfloor}$$

b. Finding the Optimal Schedule

To find the optimal schedule with the maximum number of trips based on the two parameters Y and {c_d}, we take a two-step approach. First, we determine a starting point for number of trips. This is done by calculating the absolute upper bound *P* as above and then making our first guess for the number of possible number of trips some fraction of *P*. Some experimentation has shown that a good starting guess is 4P/5. Then, we use simulated annealing to determine if it is possible to create a schedule with that number of trips. If it is, we add one to our number of trips and repeat the process until we have found the maximum possible number of trips. If it is not possible to create a schedule with trips equal to 4P/5, we subtract one from the number of trips and try again until we are able to successfully create a schedule.

c. Advantages and Disadvantages of our System

One major advantage of our system is that the method to create an optimal schedule is not overly dependent on the model of a single trip. This makes it very easy to customize. For example, if demand rises to the level where the camp director decides that it is worth it to restrict a group to only two possible campsites each night, or even just one, then only simple changes are needed in the way we define a trip, and nothing in the simulated annealing algorithm needs to change. This could also allow groups to choose a custom schedule of how far they want to go each day, rather than our current model where a group has to go (roughly) equal amounts each day, although they are allowed some variation.

The biggest disadvantage of our model is that it uses simulated annealing for something other than its intended purpose. As discussed above, simulated annealing is only guaranteed to converge if given infinite iterations, so although it is unlikely, our algorithm could potentially return a number of trips that it less than the maximum possible. Also, our simulated annealing is slow and the fact that our algorithm requires repeated simulated annealing means that it could take a long time. A mitigating factor to this is because the planning for the entire six months happens at once, the calculations only need to be done once and can be done well before the season that the river is traversable begins.

Another problem results from our model of a trip. Because we assume that a trip moves approximately 225/length miles each day, with can only vary by +-1 each day, there will be some campsites which are never used, such as the campsites which are closer to First Launch than the closest campsite we assign to an 18-day trip. This problem would not be difficult to fix with more time to develop a more nuanced model of a trip. As has been mentioned before, our simulated annealing algorithm can be used with a wide range of types of trips, and thus would not have to be changed (as long as the new model of a single trip still used its parameters to assign certain campsites to the trip on certain days).

4. Results Generated by the Current Model

a. Sample Input

We have successfully created a program that allows a given input of trips to be distributed amongst a given number of days to ensure that there no two trips ever require the same campsite. For the results discussed below, we have used the following values of input:

Quantity	Value
Total Number of Trips	120
Number of 6-day trips	36
Number of 8-day trips	24
Number of 10-day trips	22
Number of 12-day trips	19
Number of 14-day trips	9
Number of 16-day trips	6
Number of 18-day trips	4
Number of days	180
Number of campsites (Y)	50

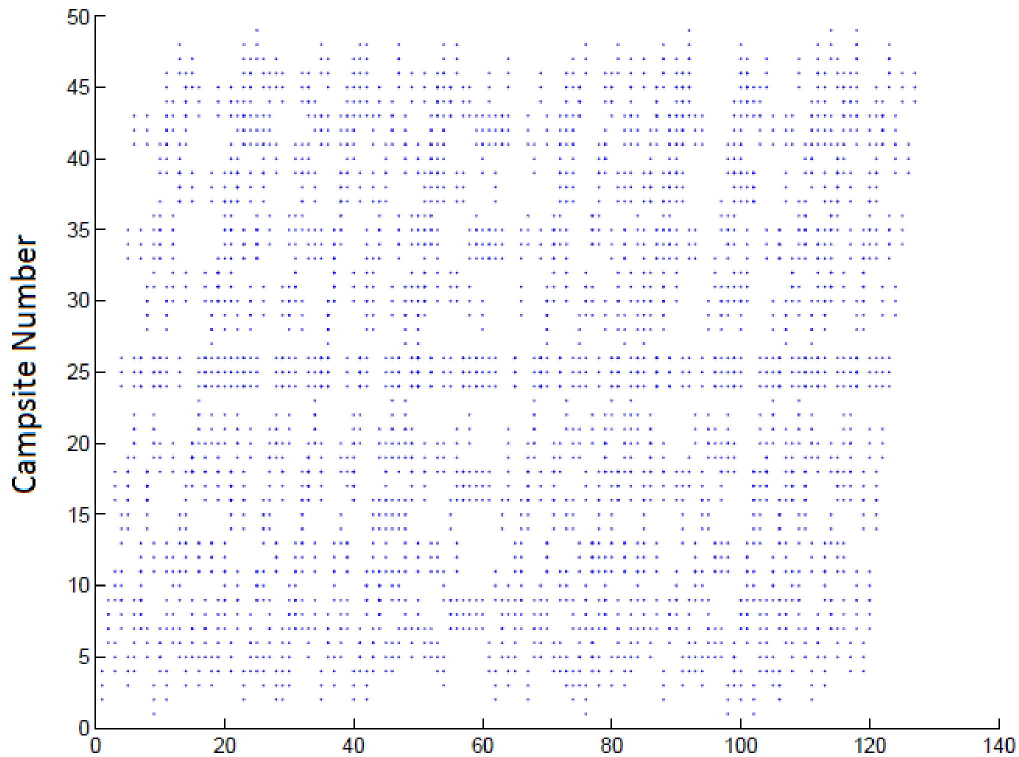
Notice that for simplicity we have only included trips of even-length. This is easy to change in the code that runs the model.

The rationale behind the distribution of days above arises by noticing that the trips that utilize motorboats (travelling 8 miles per hour) will only need to boat an average of 28.125 hours to reach the end of the river. We decided that it is reasonable to assume that riders wish to be on the boat at least three or four hours a day, meaning that most popular trips for users on motorboats are the shorter 6- or 8-day trips. Similarly, oar-powered boats travelling 4 miles per hour will require 56.25 total hours to reach the end of the river. We decided that people who use oar-powered boats are likely athletic and wish to spend more of their time boating and approximated that they will boat an average of 5 or 6 hours in a day, meaning that the most popular trips for oar-powered boats will be the 10- and 12-day trips. As mentioned before, these values are very easily altered in the program that runs the simulation.

b. Sample Output

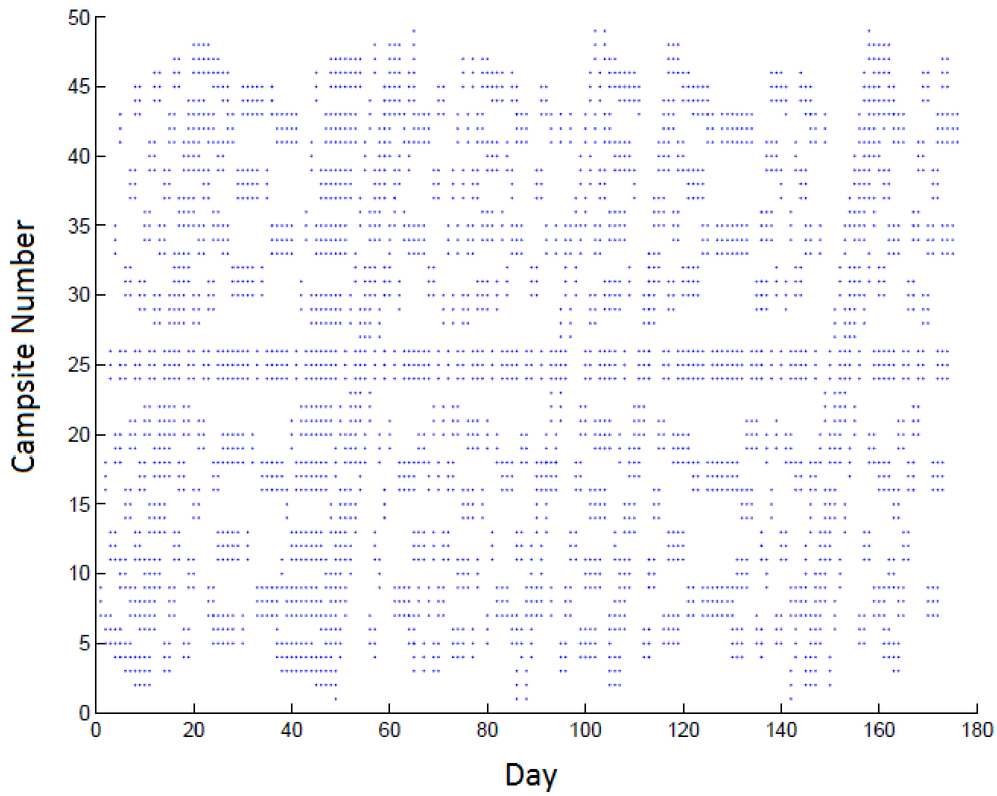
Below you can see two scatterplots: the first shows the campsites used by all of the trips after evenly distributing the trips amongst the first 120 days (before any optimizations), and the second is the result of the optimization procedure discussed in Section 3. Each point on the plot represents a campsite (y-axis) that is scheduled to be used by a trip on a specific day (x-axis). Here, dark points represent multiple trips requiring the same campsite, something that is strictly forbidden in a fully-optimized system. Plot two shows a similar plot after optimization is complete, now with zero trip intersections. These darker points are absent in the optimized model. As per the upper bound function derived in Section 3a, the upper bound on the amount of trips is 258. Below you will find a schedule determined for 120 trips. However, due to time constraints we did not let our algorithm run until completion, and so did not obtain optimal results. We believe that with time our algorithm will be able to fit many more trips into the schedule.

Figure 1: Random Distribution of trips (before optimization procedure)



Before optimization the trips are distributed evenly amongst the first 120 days. Notice that there are many dark circles – each represents an intersection of two or more trips (ie two different trips are scheduled to use the same campsite). This particular configuration has energy 1268, meaning that there are 634 intersections.

Figure 2: Trip distribution after optimization procedure



After optimization the trips are distributed so as to have no intersections. Notice that there are no dark circles and that significantly more of the 180 available days are used. The energy of this system is 0. This specific optimization required approximately 3.4 million iterations of the simulated annealing procedure, totaling approximately five minutes of computation time.

Dear River Manager:

We have prepared a method to determine the maximum number of trips which can be sent down Big Long River each year.

When deciding how to best model a trip we came to a compromise between maximizing the amount of trips and giving users many varieties of trips. We chose to enforce that each trip has a specific, pre-determined total length (determined by yourself before the start of the season). Once a trip begins, every night it has three campsites reserved for it. The boaters are told which campsites are reserved for their trip and are allowed to choose which of the three sites they wish to stop at each day. The purpose of this is two-fold. One, we allow users a decent variety of daily pace and allow them some choice in how long they stay on river each day. Two, this method ensures that no more than a third of the campsites are ever full, which will decrease the likelihood of two groups interacting and allows you, the manager, time to organize maintenance of the campsites. Although setting aside three campsites per trip, per night decreases the amount of total trips that can be sent we believe that this sacrifice is well worth the convenience gained from having non-full occupancy.

As for the specific function of our method, we ask you to input the percentage of trips that is a specific length (ie if this season you would like half of the trips to last 6 days, you would enter 50% into the slot that determines the amount of 6 day trips) and using an optimization procedure called Simulated Annealing we calculate the optimal launch schedule for a maximum amount of trips. Note that you do not need to tell us what type of boat a trip uses (oar-powered versus motorized) – we ask you only to input the length of each trip (once you have a schedule you can select which of the trips of a given length are oar-powered and which are motorized).

After some calculation, our method produces a schedule where each trip is assigned a day when it should be launched. We also produce a diagram which tells you which campsites have been reserved for each trip on each night.

Our model is currently designed to be very flexible. It is very easy to change the model so that less (or more) campsites are set aside for every trip or so that trips can have variable lengths (in this case the users, rather than yourself, would choose how long their trips would last). For this reason we hope that you do not hesitate to offer corrections or changes to the method, as it is quite likely that your suggestions will be easy to implement.

Best of luck,
COMAP Control #15421

References

[1] Aarts, Emile, and Jan Lenstra. *Local Search in Combinatorial Optimization*. London: John Wiley & Sons, 1997. 91-120. Print.

[2] Bertsimas, Dmitris, and John Tsitsiklis. "Simulated Annealing." *Statistical Science*. 8.1 (1993): 10-15. Web. 13 Feb. 2012.