Optimization of irrigation time, pipe set placements, and irrigation uniformity for a hand move system

February 6, 2006

Contents

List of Figures

1 Problem

The problem asked to generate an algorithm that would determine schedule for a hand move irrigation system that consisted of one pipe set to irrigate a 30x80 meter field. The parameters were that the schedule should involve a minimal number of moves and that the resulting application of water to the field be as uniform as possible. Furthermore, the problem stipulates that no part of the field should receive water at a rate exceeding 0.75cm per hour, and no part of the field should receive less than 2cm in a four-day irrigation circle. The pump was said to have a pressure of 420 KPa, and a flow-rate of 150 L/min. The pipe set consisted of a 10cm diameter tubes with a joint length of 20m, with 0.6cm diameter rotating spray nozzles.

2 Assumptions

1. Main Assumptions
   (a) The sprinkler has a throw radius of 13.43 meters.
   (b) Zero wind conditions While it is true that wind conditions affect precipitation uniformity,we did not explore this option due to time constraints.
   (c) The field is reasonably flat. Flatness of the field allows us to assume equal water pressure at sprinkler nozzles. Moreover, most fields are reasonably flat.
   (d) The rancher operates in 12 hour workdays. This allows for a more flexible irrigation schedule. In fact 12 hour irrigation sets are not uncommon.
   (e) All sprinklers operate on a 30 inch riser. This is the most common riser configuration that we found.
   (f) The rancher does not use the sprinklers when it rains. Rain delivers enough water to the field and has a relatively greater precipitation uniformity.
   (g) **The sprinkler application rate profile is semi-uniform for the rotating spray nozzle sprinkler.** Curiously, the only rotating spray nozzle sprinklers we found are manufactured by Nelson Irrigation. The Center for Irrigation Technology tested and listed on their website Nelson sprinkler profiles. However the largest rotating spray nozzle option for sprinklers that are used for hand move system that we found was 1/8” for the Nelson R2000WF[4]
2. Other Assumptions
   (a) There is only one accessible water source.
   (b) The boundary of the field is lined with pipes that connect to the water source.
   (c) Each pipe placement must be perpendicular to and touching one of the field boundaries.
   (d) Set up time for the pipeset will not take more than an hour.
   (e) The flow rate and water pressure from the source remains constant.

(f) As nozzle size increases so does the flow rate loss.

(g) All sprinklers operate identically and do not malfunction.

(h) The throw radius of the sprinklers may exceed the bounds of the field.

(i) No sprinkler may be placed outside the bounds of the field.

(j) At the end of the workday, the sprinklers are shut off. The pipe set need not be disassembled at the end of the workday.

3 Possible sprinkler profiles

From our research only one company makes rotating spray nozzle sprinklers, Nelson Irrigation, we obtained profiles of various sprinklers from the (CIT)Center for Irrigation Technology[2]. We did not find any sprinkler application rate profiles for a 0.6cm nozzle but we did find a profile for the Nelson Wind Fighter WF16 with a #16 Red nozzle(1/8"). The following figure from CIT shows the application rate profile for the WF16 with a #16 red nozzle at 60psi.

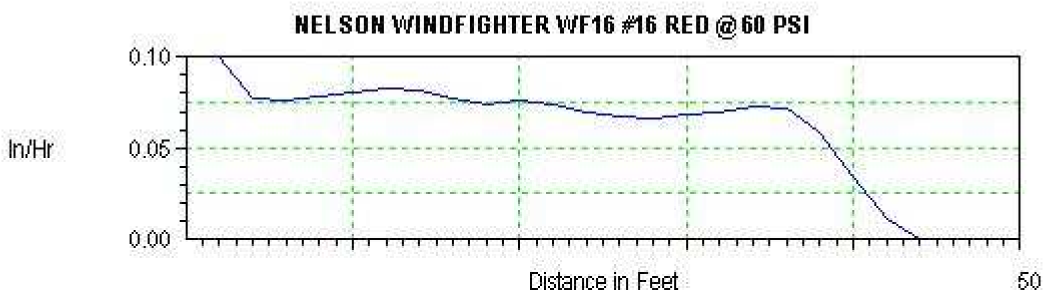

Figure 1: Sprinkler profile

We assumed that the sprinkler application rate profile for a 1/4" (approximately 0.6cm) nozzle would be similar but with an increased application rate. The flow rate for the WF16 with a 1/8" nozzle is 3.42gpm. For our nozzle diameter of 0.236" we have a flow rate of 12.57gpm so the application rate we estimate will be multiplied by a factor of 3.3 larger, accounting for an increased loss due to the sprinkler (see section 5.1.6 for the formulas used).

4 Model

4.1 Overall Approach

We employ a multi-step approach to finding an optimal solution to the problem. First, we tackle the requirement of sprinkling the entire field. Using geometrical analysis, we reduced this problem to a covering problem[3], which translates to finding the least number of equally-sized circles that can cover any given area. However, this solution results in the placement of sprinklers outside the field boundaries, and so we perturbed the solution to readjust the placement of the sprinklers, all the while maintaining the condition of complete coverage of the field. We then use this new solution as a blue print for finding the minimal number of pipe setups by experimentally ”fitting” the pipesets through the sprinklers (if possible). This is accomplished by using an algorithm that perturbs (in each iteration) the sprinkler layout, finds the minimum amount of pipe setups, perturbs the layout again (in the next iteration), finds the minimum amount of setups, and so on. After a specified amount of iterations, it then outputs the minimum of all the the found minimum setups. The rationale for perturbation is that we are willing to sacrifice some uniformity in order to find the least number of setups, while simultaneously ensuring that we are still sprinkling the entire field. We then feed this layout of pipe setups to another algorithm which generates an irrigation schedule.

4.1.1 Simulating Sprinkler Irrigation

To simulate sprinkler irrigation of the field given the sprinkler positions and the sprinkler precipitation profile and the amount of time they are on, we wrote a simple computer program in matlab. The following figures show a sprinkler profile and the output of the matlab program for that sprinkler profile with the sprinkler running for one hour. Since the sprinkler profile data is only given for discrete locations, we used a cubic spline to interpolate between sampling points for the sprinkler profile.

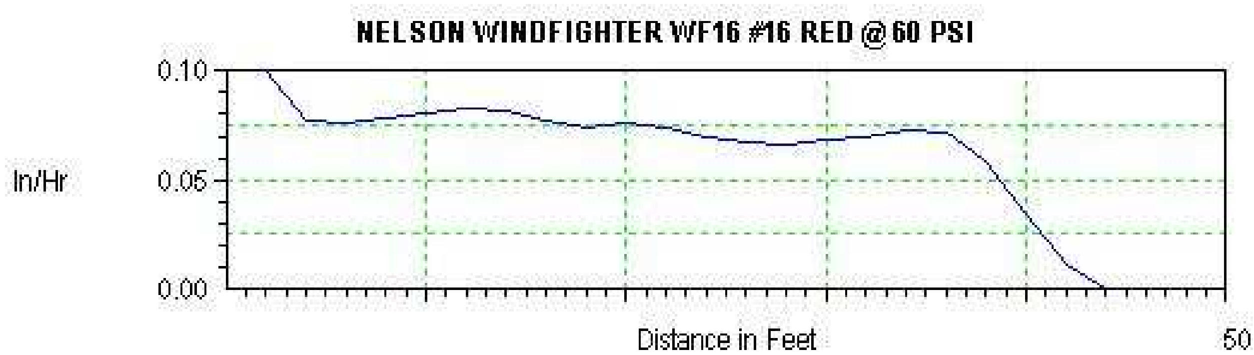

Figure 2: Example sprinkler precipitation profile

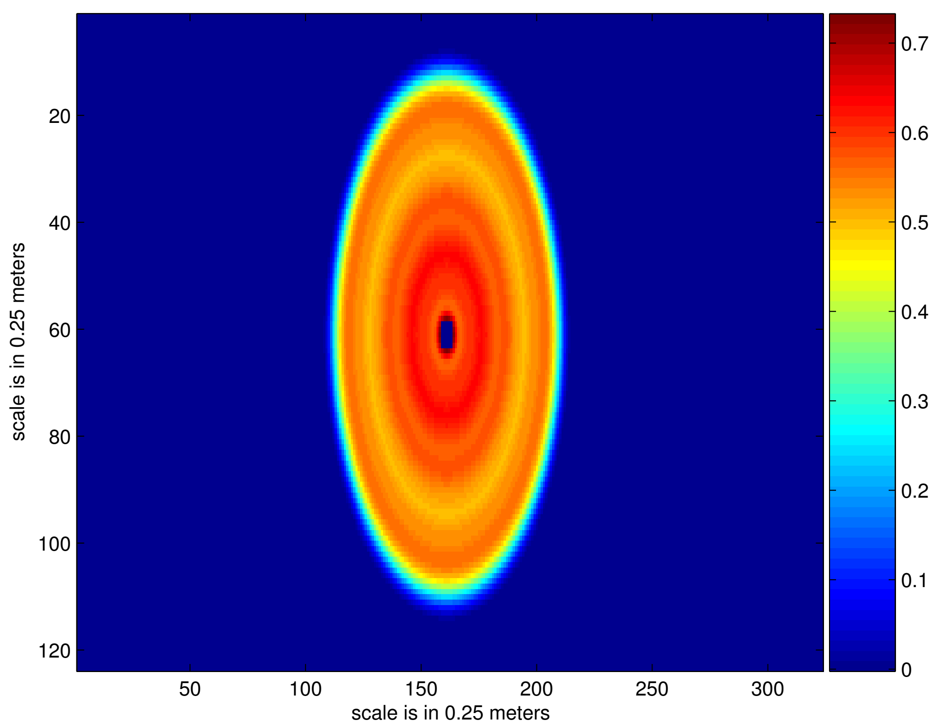

Figure 3: Matlab simulation of irrigation by sprinkler for a time of 1 hour

To represent the field in our program we used a matrix of cells. For the simulation we had a list of sprinkler positions, and for each sprinkler specified how long it runs.

1. Iterate through list of sprinkler positions.

2. For each sprinkler position call the function to simulate precipitation due to the sprinkler on the grid.

To simulate precipitation from a sprinkler we used a simple nested for loop to iterate through the cells that are within the whetted radius of the sprinkler. For each of these cells we computed the distance from a cell to the sprinkler and then used the given sprinkler precipitation rate profile and the length of time the sprinkler runs to calculate the amount precipitation received by that cell. This number is then simply added to the current precipitation amount in the cell.

4.1.2 Complete Sprinkler Coverage of Field

The problem stipulates that no area of the field may receive less than 2 cm of water every 4 days. This implies that it is necessary for every part of the field to be watered. To do this, we first think of a sprinkler's whetted area as a circle located in within a rectangle which represents the field. The problem then reduces to completely covering this rectangle with circles. However, because the distribution profile for the sprinkler is near uniform (see fig 2), allowing for radial overlaps will disturb overall uniformity and increase the number of sprinklers needed to cover the field. Hence, it is best to minimize overlap by minimizing the number of sprinklers while ensuring every part of the field is completely covered. This can, however, be restated as a covering problem, in which we find the smallest number of equally-sized circles that can cover a given plane. Fig. 3 represents this solution. It can be shown that no other configuration of circles can cover this plane more efficiently than this method[3].

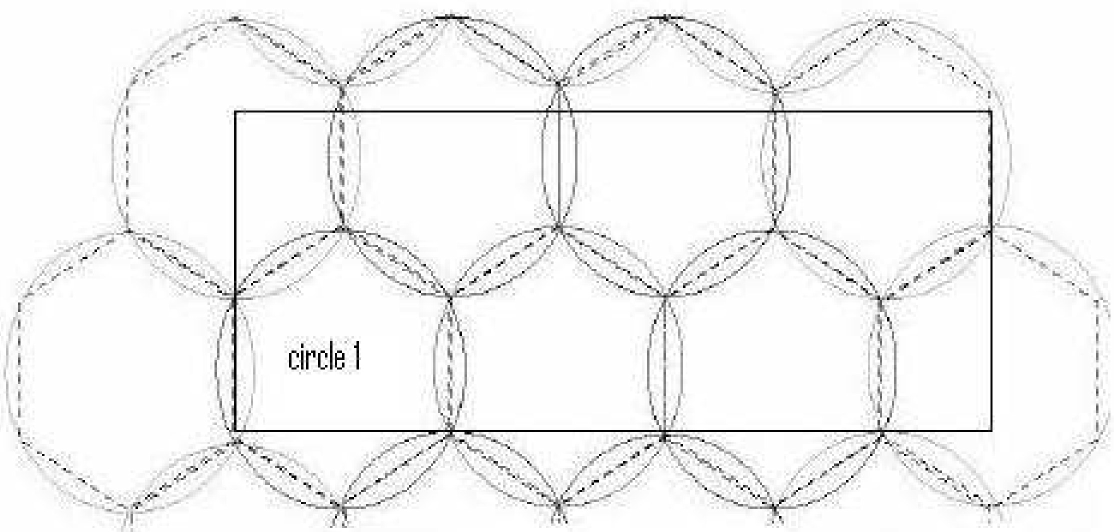

Figure 4: hexagonal covering

4.1.3 Adjustment of the Covering Problem Solution

The solution presented above, however, is not completely useful to us, since it would result in some sprinklers being positioned outside the field's boundaries. Moreover, as it turns out, adjusting can also lead to an increase in uniformity. Hence, we shift the sprinkler positions not only to ensure that all of them remain within bounds, but to perhaps maximize the overall precipitation uniformity (we shall see in the next section that continuous shifting in this manner actually yields an optimum configuration). In the figure 4 we show a possible placement of the field with the solution of the covering problem:

Thus, we need to adjust this solution to ensure that the sprinklers are all within the field, while maintaining the condition that the field is still covered as uniformly as possible. From the hexagonal covering pattern (see fig 3) we know that the center of circle 1 is supposed to be located at a distance of $X = R\sqrt{3}/2$ to the right of field's left boundary. From the spline interpolation of the data from the

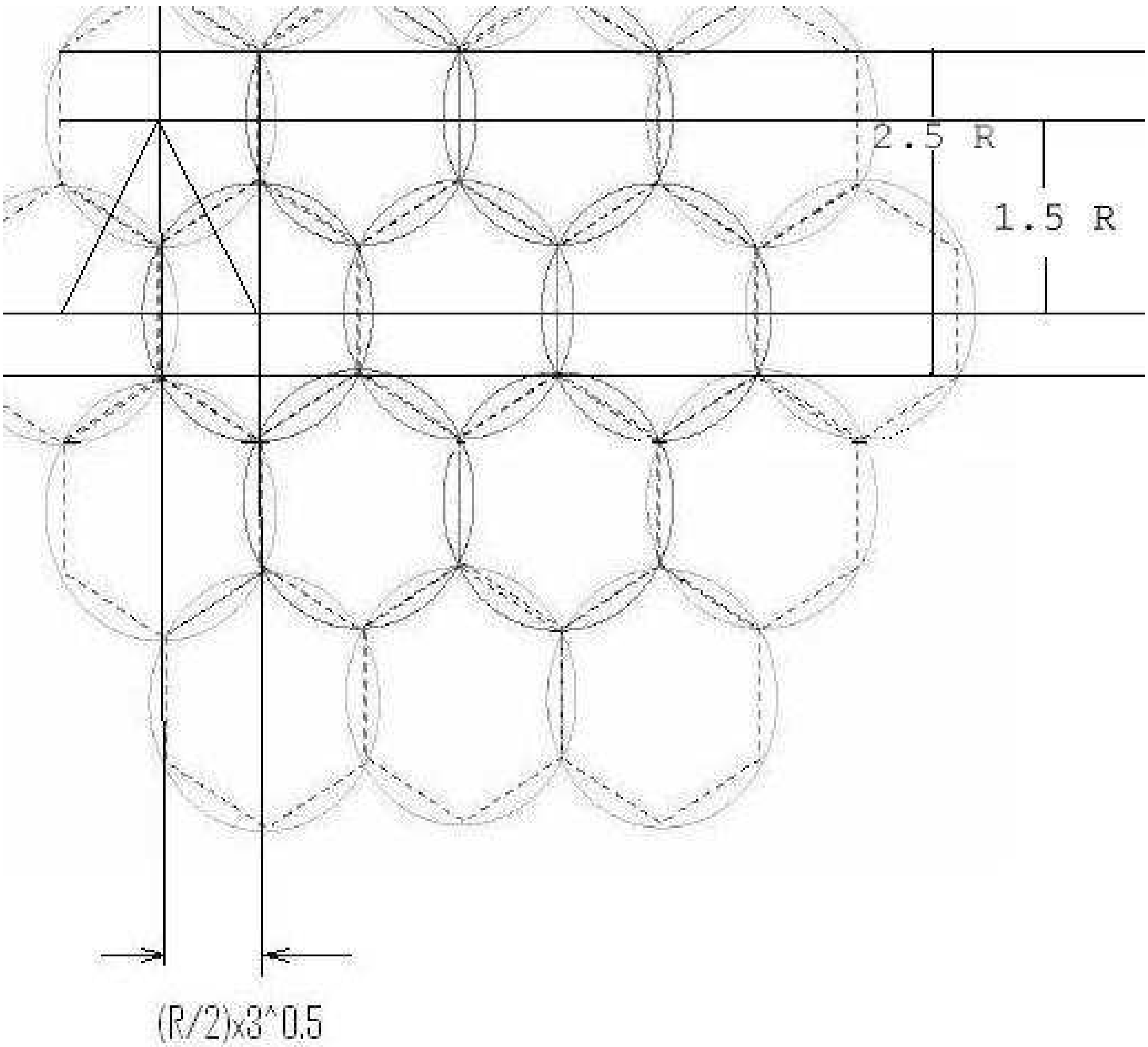

Figure 5: hexagonal covering

sprinkler coverage profile R was estimated to be 13.43 meters. But at 13.43 meters the precipitation rate is zero. We not only want the rate greater than zero but also for the total application to reach 2 cm in 5 hours. Placing this constraint on the precipitation rate, we find that it has the radius of only 11.5 meters and anything beyond this will not provide the required coverage. Thus, to achieve the most complete coverage we need to calculate the x-coordinate of circle 1, using R=11.5 meters, and then place all the other circles on the left row with the regular spacing of 13.43 meters. This will result in the shift of the left row 13.43 meters-11.5 meters=1.93 meters to the left, thus not only keeping the sprinklers within our bounds but also increasing overall precipitation uniformity.

4.1.4 Maximizing Precipitation Uniformity

Applying water to the field as uniformly as possible, is also a main concern, since this leads to efficient crop yields[6]. We tackled this optimization problem using a Simultaneous Perturbation Stochastic Approximation (SPSA) optimization algorithm[5], in which we minimized the standard deviation from the average overall precipitation in the field. The code of this algorithm is included in the appendix for the interested reader. Below are figures showing the result of the SPSA algorithm after a given number of iterations.

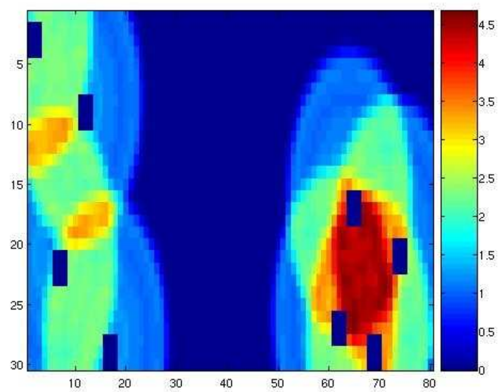

Figure 6: Random Initial sprinkler placement

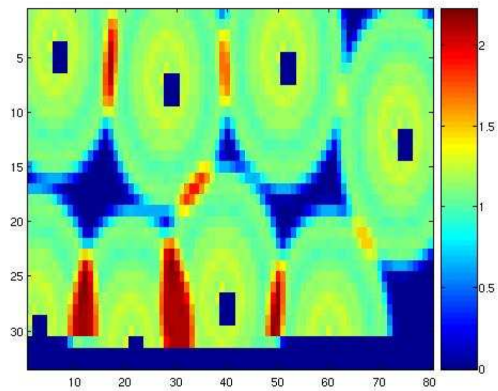

Figure 7: Sprinkler placement after 100 iterations

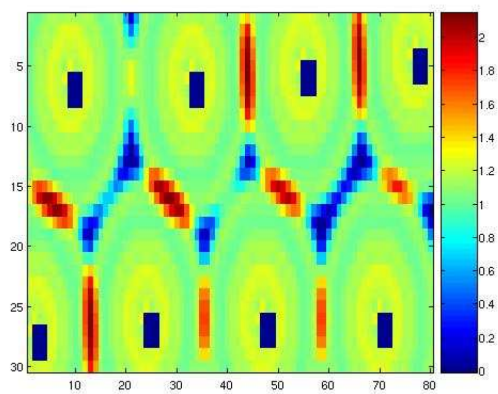

Figure 8: Sprinkler placement after 500

Interestingly enough, after 5,000 iterations, the method seems to mimic the shifted covering problem solution. That is, after 5,000 iterations, the maximum uniformity is achieved by positioning the sprinklers in a manner similar to the shifted covering problem solution! Unfortunately, due to lack of access to more powerful computing machines, we were unable to obtain the sprinkler SPSA coordinate outputs that correspond to the 5,000th iteration, but since the methods are similar, it is reasonable to assume that our shifting method achieves a uniformity coefficient that adequately approximates the one produced by the SPSA. Hence we can use our method as a representation of the sprinkler output that maximizes uniformity.

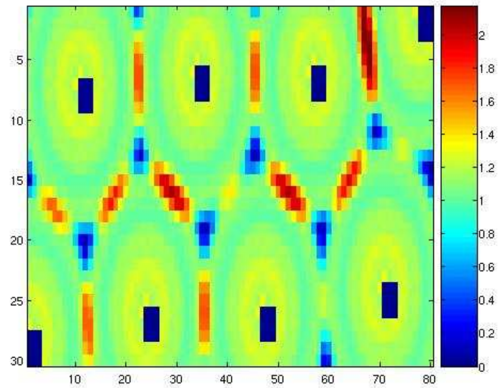

Figure 9: Sprinkler placement after 5000 iterations

4.1.5 Algorithm: Minimization of Pipe Setups

This algorithm takes as input the coordinates of sprinklers positions as determined by our approximation to the SPSA solution. From this layout, it then minimizes the number of pipe setups by first selecting a sprinkler closest to the upper corner of the field. It then calculates the distances of all other sprinklers from it, and selects the sprinkler with the shortest LATERAL distance (diagonal distances will be ignored because we assume we cannot place a pipe diagonally on the field). At this point, it then evaluates whether or not this distance is less than or equal to the pipe length (20 meters). If it's less than this length, it then calculates the precipitation rates of points located within the overlapping radii. If this rate exceeds .75 cm/hr, it goes onto the next closest sprinkler.

4.1.6 Irrigation system calculations

The problem specifies that the mainline pipe used in the hand move system is aluminum and has a diameter of 10cm. In addition the sprinkler nozzle size is 0.6cm. The water source has a pressure of 420Pascals and a possible flow rate of 150 L/min. For our calculations we used the following formulas from Rainbird[2].

Hazen-Williams

$$\text{Pressure Loss (psi)} = 4.55\frac{(\frac{Q}{C})^{1.852}}{ID^{4.87}}L$$

$$\begin{aligned}
Q &= \text{Pipe flow (gpm)} \\
C &= \text{Roughness coefficient(Aluminum w/ couplers} = 120 \\
ID &= \text{Pipe inside diameter (in)} \\
L &= \text{Pipe length (ft)}
\end{aligned}$$

Nozzle Discharge

$$\text{Discharge (gpm)} = 29.82\sqrt{P}D^2C_d$$

$$\begin{aligned} P &= \text{Nozzle pressure (psi)} \\ D &= \text{Nozzle orifice diameter (in)} \\ C_d &= \text{Nozzle discharge coefficient(tapered} \simeq .96 \text{ or } .98) \end{aligned}$$

There are 0.145 kiloPascal per psi so the system pressure is at most 60.9(psi). The nozzle size is $0.6/2.54 = 0.236$(in) and assuming a Nozzle discharge coefficient of 0.97 we obtain a the flow rate per sprinkler of 12.57(gpm)= 47.58L/m. The pressure loss due to the mainline pipe assuming four sprinkler is 0.0122(psi). Thus it can be neglected in our calculations. We assume that each sprinkler is on a 30in riser and the riser is a 1(in) diameter steel pipe. The pressure loss assuming a flow of 12.57(gpm) is 0.058(psi). Thus we can also neglect pressure loss due to the riser.

5 Results

Our model generated the following optimal pipeset configuration:

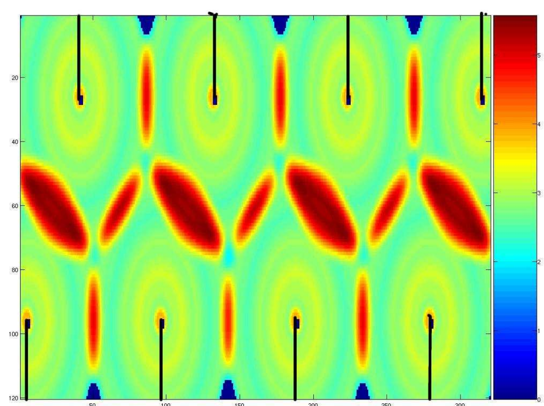

Figure 10: Best sprinkler placement, with pipe sets(shown as black lines).

This configuration consists of 8 pipe movements each in intervals of 5 hours (with an assumed 1 hour time for moving and set up of equipment) and results in a total irrigation time of 48 hours every 4 days, or approximately 12 hours per day. As can be seen from the graph above, each pipeset contains only one sprinkler. The reason for this is that if the sprinklers were placed any closer than 21 meters from each other, the overlap of their whetted areas will yield a precipitation rate greater than 75 cm per hour, which violates one of our constraints. Approximately less than 1% of the field, however, did not receive a sufficient amount (2 cm) of water. This is not a major concern, since these are the areas located at the edges, where the crops could easily be damaged by a myriad of factors, such as bypassing farm equipment, lower soil quality.

The table below is the generated irrigation schedule for the repositioning of the sprinklers, given a 12-hour work day for a rancher. Each pipe is set in place for 5 hours.

Irrigation schedule		
Day	Time	Sprinkler position (x position, y position) in meters from the lower left corner of the field
Day 1	8am	1, 23.75
Day 1	1pm	23.75, 23.75 (move the sprinkler and pipe set to new location)
Day 1	6pm	Turn off sprinkler to avoid over irrigation.
Day 2	8am	46.75, 23.75
Day 2	1pm	69.5, 23.75
Day 2	6pm	Turn off sprinkler to avoid over irrigation.
Day 3	8am	10, 6.25
Day 3	1pm	32.75, 6.25
Day 3	6pm	Turn off sprinkler to avoid over irrigation.
Day 4	8am	55.5, 6.25
Day 4	1pm	78.25, 6.25
Day 4	6pm	Turn off sprinkler to avoid over irrigation.
Day 5		Repeat schedule.

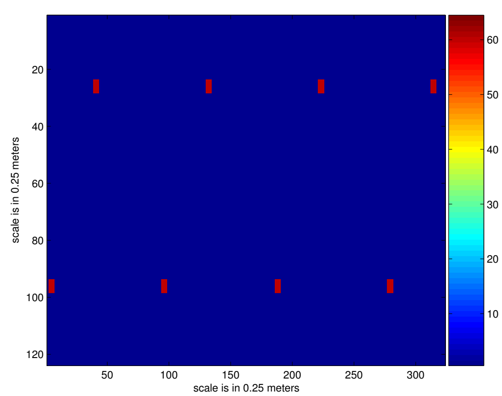

Figure 11: Optimal sprinkler placement

Based on our assumptions and the design of our algorithm, there is no faster way to irrigate this field while maintaining such a high measure of precipitation uniformity.

As expected, the above data is consistent with all the earlier analysis, given our assumed sprinkler distribution profile. The distributions were, of course, uniform as we predicted. In fact, our solution produced a uniformity coefficient of 0.89, which is unsurprisingly close to the optimal value of .90 that was generated by the SPSA after 5000 iterations.

5.1 Weaknesses

1.The model assumes no-wind conditions and does not account for the change in sprinklers' profile due to the wind. In the wind conditions the radius of the sprinkler facing the wind would shorten up, and th one along the direction of the wind would elongate thus changing the distribution of precipitation rate and uniformity. Such an alteration in a precipitation profile could lead to a completely different model. 2.SPSA ideally could have produced the best possible solution, but we did not have time to run it for enough number of iterations. The ideal solution would have been achieved if the simpler programming language would have been used. Using FORTRAN would increase the speed of calculations by about 10000 times. 3.The rancher has to operate on a twelve-hour a day schedule, instead of an eight-hour one. It means a work day longer than a regular one. 4.The rotating spray nozzle profile we used in our model represents the nozzle of the size twice as small as the one described in the problem. Even though we scaled the precipitation rate using provided in the problem flow rate and pressure, there is no guaranty that the sprinkler profile would not change with flow rate. Ideally, if had the necessary equipment, we could have gone out and measured the precipitation rate as a function as distance from the sprinkler for a given spray nozzle. The solution would have been a much better representation of a reality. 5.Coefficient of Uniformity(CU) is an average value. It provides an average deviation from the mean coverage. Different sprinkler patterns may return the same CU. So if the area is over watered due to the overlap of different sprinklers placed at different times, or over watered in one spot and under watered at another the CU may come out to be the same. (http://cati.csufresno.edu/cit/rese/88/880106). Thus it is not the most accurate way to measure the uniformity and water application; especially for our purposes, since we care less about minor over watering than about under watering. 6.We were not able to validate our model in real-life conditions. All the verification was done using computer simulations. In reality the distribution may not work out exactly the way we assume it would due to various natural conditions (wind), field location (hills), or sprinklers' imperfections and deviations from the provided profile.

5.2 Further Improvements

To make the model more realistic, we should include various weather conditions, possible wind, rain, topography. Make the algorithms more general by taking into account the possibility of other sprinkler precipitation rate profiles, nozzle size, and sprinkler types.

References

[1] Center for Irrigation Technology California State University Fresno 5370 N. Chestnut Ave. Fresno, CA 93740-0018 Phone: (559) 278-2066 Fax: (559) 278-6033 Internet: http://cati.csufresno.edu/

[2] Rainbird Reference formulas http://www.rainbird.com/pdf/ag/FormulasUS.pdf

[3] R. Kershner, ”The number of circles covering a set”, American Journal of Mathematics, 61:665–671, 1939.

[4] Nelson R2000WF Rotator Nozzle Options URL:chttp://www.nelsonirrigation.com/data/products/ACF25.pdf

[5] ”Simultaneous Perturbation Stochastic Approximation” URL:http://www.jhuapl.edu/SPSA/

[6] Lamm, Freddie R., ”Uniform Application by Center Pivot Sprinklers”, URL: ”http://www.oznet.ksu.edu/irrigate/UICCP98.html

A Program Code

A.1 main.m

```
positions=[40 15;7 0; 14 0];%row of sprinklers

cell_size=0.25;%conversion to factor for cell size to meters

[pp_sprinkler_distribution]=init_sprinkler_distribution(cell_size)
num_sprinklers=[1 1 1 1 1 1];% go vertical up or down, or horizontal left or right,
%0 is upwards vertical, 1 is downwards vertical, 2 is rightwards horizontal, 3 is leftwords horizo
sprinkler_spacings=[0;0;0;0;0;0];
time=[100 100 100];%time to run each sprinkler in hours
orientations=[2 2];
grid=run_schedule(positions,orientations,num_sprinklers,sprinkler_spacings,time,cell_size);
grid=plot_sprinklers(grid,positions,orientations,num_sprinklers,sprinkler_spacings,time,cell_size)


grid=grid';%transpose the matrix so we have the width on the x-axis and the height on the y-axis.
uniformity=1-std2(grid)/(1+std2(grid'))%measure of uniformity using the standard deviation of the
%interval [0,1].
plot_field(grid);
```

A.2 init_sprinkler_distribution.m

```
function [pp_sprinkler_distribution]=init_sprinkler_distribution(cell_size)
global pp_sprinkler_distribution;
global sprinkler_radius;
x=0.305.*[0    1    2     4     6       8        10    12      14      16       18       20      22
y=2.54*3.*[ 0.1 0.1  0.1  0.077  0.076   0.079    0.08  0.083   0.081   0.077   0.074   0.076    0
        sprinkler_radius=14/cell_size;%maximum radius of circle that sprinkler soaks in meters
pp_sprinkler_distribution=spline(x,y);
\end{document}
```

A.3 run_schedule.m

```
%Orientation tells us if the sprinklers go vertical up or down, or horizontal left or right,
%0 is upwards vertical, 1 is downwards vertical, 2 is rightwards horizontal, 3 is leftwords horizo
function [grid]=run_schedule(positions,orientations,the_num_sprinklers,the_sprinkler_spacings,time
grid=new_grid([81/cell_size 31/cell_size]);%Create a grid for the field where each grid cell is 1
num_moves=length(times);
for i=1:num_moves


time=times(1,i);%time in hours to run the sprinklers.
num_sprinklers=the_num_sprinklers(1,i);
position=positions(i,:);
orientation=orientations(1,i);
sprinkler_spacing=the_sprinkler_spacings(i,:);%position of the sprinklers on the pipe in meters.
water_pressure=414;%water pressure in kilPascal
sprinklers=new_sprinklers(position,num_sprinklers,orientation,sprinkler_spacing,cell_size);
%pipe_set=new_pipe_set(sprinklers,5,)
grid=sprinkle_field(grid,sprinklers,water_pressure,time,cell_size);
%plot_field(grid);
%pause(2);
%hold off;
end
```

A.4 new_sprinklers.m

```
%Orientation tells us if the sprinklers go vertical up or down, or horizontal left or right,
%0 is upwards vertical, 1 is downwards vertical, 2 is rightwards horizontal, 3 is leftwords horizo

function [sprinklers]=new_sprinklers(position,num_sprinklers,orientation,sprinkler_spacing,cell_si
sprinklers=zeros(num_sprinklers,2);
if orientation==0
for i=1:num_sprinklers
sprinklers(i,:)=round(position./cell_size+[1 1]+1/cell_size.*[0 double(sprinkler_spacing(i))]);%th
end
end
if orientation==1

for i=1:num_sprinklers
sprinklers(i,:)=round(position./cell_size+[1 1]+1/cell_size.*[0 -1*sprinkler_spacing(i)]);
end
```

```
end
if orientation==2

for i=1:num_sprinklers
sprinklers(i,:)=round(position./cell_size+[1 1]+1/cell_size.*[1*sprinkler_spacing(i) 0]);
end
end
if orientation==3

for i=1:num_sprinklers
sprinklers(i,:)=round(position./cell_size+[1 1]+1/cell_size.*[-sprinkler_spacing(i) 0]);
end
end
end
```

A.5 sprinkle_field.m

```
function [grid]=sprinkle_field(grid,sprinklers,water_pressure,time,cell_size)
s_sprinklers=size(sprinklers);
num_sprinklers=s_sprinklers(1,1);
for i=1:num_sprinklers
grid=run_sprinkler(grid,sprinklers(i,:),water_pressure,time,cell_size);
end
```

A.6 run_sprinkler.m

```
%sprinkler_parameters is a vector that contains the following list of informations
%sprinkler_position,sprinkler_nozzile_radius,sprinkler_height
function [grid]=run_sprinkler(grid,sprinkler_position,water_pressure,time,cell_size)
global sprinkler_radius;
s_grid=size(grid);
size_x=s_grid(1,1);
size_y=s_grid(1,2);

%bounding_box=new_bounding_box(sprinkler_pos,sprinkler_radii)

%sprinkler_flow=19;%sprinkler flow rate in L/min
pos=sprinkler_position;
area=3.14159265359*sprinkler_radius^2;
for i=pos(1,1)-sprinkler_radius:pos(1,1)+sprinkler_radius
for j=pos(1,2)-sprinkler_radius:pos(1,2)+sprinkler_radius
if i<1|i>size_x|j<1|j>size_y
```

```
%print 'continuing';
continue;
end
if norm([i j]-pos)<=sprinkler_radius
distance=norm([i j]-pos);
%[i j]

grid(i,j)=grid(i,j)+time*sprinkler_distribution(distance*cell_size);%(60.0*sprinkler_flow)/area;
end
end
end
end
```

A.7 plot_sprinklers.m

```
function [grid]=plot_sprinklers(grid,the_positions,orientations,the_num_sprinklers,the_sprinkler_s
num_moves=length(times);
for i=1:num_moves


time=times(1,i);%time in hours to run the sprinklers.
num_sprinklers=the_num_sprinklers(1,i);
position=the_positions(i,:);
orientation=orientations(1,i);
sprinkler_spacing=the_sprinkler_spacings(i,:);%position of the sprinklers on the pipe in meters.

sprinklers=new_sprinklers(position,num_sprinklers,orientation,sprinkler_spacing,cell_size);

s_sprinklers=size(sprinklers);
num_sprinklers=s_sprinklers(1,1);
g_size=size(grid);
grid_x=g_size(1,1);
grid_y=g_size(1,2);
for i=1:num_sprinklers

sprinkler=sprinklers(i,:);
for i=round(-round(1/cell_size)/2):round(round(1/cell_size)/2)
for j=round(-round(1/cell_size)/2):round(round(1/cell_size)/2)
if sprinkler(1,1)+i>0&&sprinkler(1,1)+i<=grid_x&&sprinkler(1,2)+j>0&&sprinkler(1,2)+j<=grid_y

grid(sprinkler(1,1)+i,sprinkler(1,2)+j)=0;%make a dot so we can see the sprinkler on the plot.
end
```

```
end
end
end
end


end
```

A.8 plot_field.m

```
function []=plot_grid(grid)
gsize=size(grid);
size_x=gsize(1,1)
size_y=gsize(1,2)
imagesc(grid);
colorbar;
axis manual;
axis([1 size_y 1 size_x]);
xlabel('scale is in cell_size meters');
ylabel('scale is in cell_size meters');
%axis manual;
%axis([1 80 1 30]);
%plot(grid);
%grid=grid';
%[x, y] = meshgrid([1:1:size_x],[1:1:size_y]);
%surf(x,y,grid);
%hold off;
%contour(grid);


end
```

B SPSA algorithm

B.1 spsa.m

```
%function pos=optimize(pozitions)



theta=[0 1; 5 20; 10 7; 60 25; 63 15; 67 27; 15 27; 72 19];
```

```
main(theta)


alpha=0.2;
gamma=0.101;
a=5;
c=5;
A=50;
p=size(theta);
n=7000;

thetamax=[80 30;80 30;80 30;80 30; 80 30; 80 30;80 30;80 30];
thetamin=[0 0; 0  0; 0  0; 0 0;0 0; 0 0; 0 0; 0 0];

for k=0:n-1
ak=a/(k+1+A)^alpha;
ck=c/(k+1)^gamma;
delta=2*round(rand(p(1),p(2)))-1;
thetaplus=max(min((theta+ck*delta), thetamax),thetamin);
thetaminus=min(max((theta-ck*delta), thetamin),thetamax);
yplus=main2(thetaplus);
yminus=main2(thetaminus);
ghat=30*(yplus-yminus)./(2*ck*delta);
theta=theta-ak*ghat;
theta=min(theta, thetamax);
theta=max(theta, thetamin);
end
theta

figure(2)
main(round(theta))
```

B.2 main2.m

```
function uniformity=main(positions)
%positions=[2 20;50 10; 24.5 24.25; 33.75 5.75;47 24.25; 56.25 5.75; 69.5 24.25; 78.75 5.75];
%impact nozle
%[0 27; 30 27; 16.5 2; 49.5 2; 62 27; 82.25 2];
%rotating spray nozle positions

%Radius spacing of spry nozzle
%%[7 30; 35 30; 63 30; 0 20; 28 20; 56 20;    7 10; 35 10; 63 10;    0 0; 28 0; 56 0];%
```

```
cell_size=1;%conversion to factor for cell size to meters

[pp_sprinkler_distribution]=init_sprinkler_distribution(cell_size);
orientations=[0 0 0 0 0 0 0 0];%Orientation tells us if the sprinklers go vertical up or down, or
%0 is upwards vertical, 1 is downwards vertical, 2 is rightwards horizontal, 3 is leftwords horizo
num_sprinklers=[1 1 1 1 1 1 1 1];
sprinkler_spacings=[0 ;0;0;0;0;0;0;0];
time=[6 6 6 6 6 6 6 6];
grid=zeros(80/cell_size,30/cell_size);
grid=run_schedule(positions,orientations,num_sprinklers,sprinkler_spacings,time,cell_size);
disp('plotting sprinklers');
grid=plot_sprinklers(grid,positions,orientations,num_sprinklers,sprinkler_spacings,time,cell_size)


grid=grid';%transpose the matrix so we have the width on the x-axis and the height on the y-axis.
uniformity=1-std2(grid)/(1+std2(grid'))%measure of uniformity using the standard deviation of the
%interval [0,1].
plot_field(grid);
```

B.3 perturbation_algorithm.m

```
function []=perturbation_algorithm(sprinkler_positions,delta,pipe_set_length,field_width,field_hei
s_sprinklers=size(sprinkler_purterbations);
num_sprinklers=s_srinklers(1,2);
pipe_positions=[];
sprinkler_positions=sortrows(sprinklers_positions);%Sort the order of the sprinkler positions, so
purterbed_sprinkler_positions=[];
pipe_positions=[];
pipe_orientations=[]
for i=1:num_sprinklers
%Start search at the sprinkler that is closest to the lower left hand corner first. And then try t
sprinkler=sprinkler_positions(i,:);
if sprinkler(1,1)<sprinkler(1,2)%If the sprinkler is closer to the x-axis than the y-axis than we
if abs(sprinkler(1,2)-field_height)<abs(sprinkler(1,2)-0)%if the sprinkler is closer to the top of
new_pipe_position=[sprinkler(1,1) field_height];
pipe_orientations=[pipe_orientations 1];%pipe orientation is vertical downwards.
else
new_pipe_position=[sprinkler(1,1) 0];%pipe orientation is vertical upwards
end
```

```
else
if abs(sprinkler(1,1)-field_width)<abs(sprinkler(1,2)-0)%if the sprinkler is closer to the right s
new_pipe_position=[sprinkler(1,1) field_height];
pipe_orientations=[pipe_orientations 2];%pipe orientation is leftwords horizontal
else
new_pipe_position=[sprinkler(1,1) 0];

pipe_orientations=[pipe_orientations 3];%pipe orientation is rightwords horizontal
end

end
pipe_positions=[pipe_positions;new_pipe_position];
for j=1:num_sprinklers
if j~i
pos1=sprinkler_posistions(i,:);
pos2=sprinkler_posistions(j,:);
if norm(pos1(1,1)-pos2(1,1))<norm(pos1(1,2)-pos2(1,2))
if norm(pos1(1,1)-pos2(1,1))<pipe_set_length+delta

end
else
if norm(pos1(1,2)-pos2(1,2))<pipe_set_length+delta

end
end
end
end
end
end
```