AURORA DEL CAMP – SPECIFICATION

JULIAN PFEIFLE

This document outlines some of the issues that arise in the implementation of Gilad's proposal [1].

Given the availability of powerful free and open source solvers for integer programs such as CBC [2], it seems natural to pursue an integer programming formulation. Of course, free solvers are not as good as the best commercial ones, but the most recent benchmarks [3] indicate that CBC is reasonably competitive; more precisely, it's the most competitive among all solvers that have an open source license (in the case of CBC, the "Eclipse Public License") that permits Gilad to use it commercially without paying any license fees.

1. Problem formulation

We start by translating Gilad's document into this language.

1.1. **Variables.** We work with a set C of crops, the set $W = \{1, \dots, 52\}$ of weeks in the year, and a set A of "unit acres", i.e., indivisible units of farmland dedicated to a single crop or task. The set A has a distinguished member $a_0 \in A$ that stands for off-field work. In this fictitious area of acre, any number of tasks may be performed simultaneously, while only one thing at a time may be done in all acres $A \setminus a_0$.

We will have several different families of variables, one family for each specific task: ti for tilling, rv for rotovating, gm for green manure planting, ft for fertilizing, bb for bed building, si for setting up irrigation, sr for setting rows, ss for soaking seeds, cs for cutting or separating cloned seeds, gc for false germination and cleaning, pl for planting, ha for harvesting.[1]

Each of these families contains variables indexed by the type of crop $c \in C$, the week $w \in W$ of the year, and the piece of land $a \in A$. Each of the variables can take on the value 0 or 1, depending on whether or not the task at hand is undertaken for crop c in the unit acre a during week w. For instance, the set of seed-soaking variables is $ss = \{ss_{c,w,a} : c \in C, w \in W, a \in A\}$. Some families have less variables: for instance, the tilling variables are just $ti = \{ti_{w,a} : w \in W, a \in A\}$.

Now finish enumerating the variables, and count how many there are.

1.2. **Constraints.** Some of the more obvious constraints are the following:

In each week, there is only one crop at each unit acre:

$$\sum_{c \in C} x_{c,w,a} = 1 \qquad \text{for all } w \in W \text{ and } a \in A$$

Finish the constraints, starting with the precedence constraints

Date: Version of November 16, 2011.

[1] **I left out buying seeds, because that seems easy enough to do. Is that all right?**

1.3. **Objective function.** Each crop $c \in C$ has a yield of $y_{c,w}$, depending on the week $w \in W$ it is planted. Since the default direction in optimization appears to be minimization, we make the yields negative.

The objective function is thus

$$f = -\sum_{c \in C, y \in Y} y_{c,w} \sum_{a \in A} x_{c,w,a}$$

2. Server-side technology

Gilad's intention is to make the program available on a server. That's fine, except that we need to be able to install c++ and cbc on such a server.

References

[1] G. Buzi, *Aurora Del Camp Crop Planner — a small farm plans big*, November 2011.

[2] *Cbc (coin-or branch and cut), an open-source mixed integer programming solver written in C++*. `https://projects.coin-or.org/Cbc`.

[3] H. D. Mittelmann, *Performance of optimization software — an update*. `http://plato.asu.edu/talks/mittelmann_bench.pdf`, November 2011.