.

AURORA DEL CAMP – SPECIFICATION

JULIAN PFEIFLE

This document outlines some of the issues that arise in the implementation of Gilad's proposal [2].

Given the availability of powerful free and open source solvers for integer programs such as CBC [3], 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 [4] 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. Event-based modeling

1.1. **Tasks to be considered.** These come in two types: $T = T_c \cup T_s$, where the tasks in T_c only affect the field, and are thus common to all crops, and the tasks T_s are specific to each crop. They are:

Tasks common to all crops: 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, we for weeding:

$$T_c = \{ti,\ rv,\ gm,\ ft,\ bb,\ si,\ sr,\ we\}$$

Tasks specific to a crop: by for buying seeds, ss for soaking seeds, cs for cutting or separating cloned seeds, gc for false germination and cleaning, pl for planting[1], fu for fumigating, th for thinning, tr for trimming, co for covering, ha for harvesting:

$$T_s = \{by,\ ss,\ cs,\ gc,\ pl,\ fu,\ th,\ tr,\ co,\ ha\}$$

We record the precedence constraints between these tasks in a directed acyclic graph E. Thus, $t_1 \longrightarrow t_2$ (also written $t_1 < t_2$) is a directed edge in E if t_1 must be completed before t_2 can start.

1.2. **Overview of the model.** We separate the *scheduling* part of the problem, in which the appropriate sequencing of events is determined, from the *allocation* part, in which tasks and crops are assigned their proper place in the field. First, the optimal sequencing of events is determined in a way that respects the available field space and work force using an *event-based* formulation (see below); in a second step, each task is allocated its proper space in the field. Separating the two phases makes it easier to formulate each one, and presumably makes them (and thus, the whole problem) easier to solve.

In general terms, we will plan over several years. In the final program, there will be an interface to put new tasks into a *task queue* (a set of events that is not scheduled yet), and an interface to record the actual progress of tasks. This act of recording the real start and end time of activities, environmental changes, changes in the available work force, etc., sets certain variables in the problem formulation to fixed, known values, and allows for frequent new optimization runs that dynamically take into account the latest developments.

Date: Version of November 23, 2011.

[1] We consider transplanting and planting to be the same process.

1.3. **Scheduling.** Here we follow [1]. Put briefly, the main *non-renewable* resource consumed is "time", while some of the *renewable* ones are "space in the fields", "money" and "gasoline". Money is put back into the system by selling the crops, and field space by tilling the remains of the crop. Gasoline is renewable, but costs money.

To facilitate the modelling process, we maintain the same language as [1] and use the word *task* to refer to an element of $T = T_c \cup T_s$ as above, and the word *activity* to refer to a task $t \in T$ executed on a specific unit lot $\ell \in L$. Thus, the set of activities is

$$A \;=\; \big\{a_{t,\ell} : t \in T,\ \ell \in L\big\}.$$

The idea is to have "planting a batch of tomatoes in summmer 2011", "planting another batch of tomatoes in summer 2011", and "planting a third batch of tomatoes, but in the summer of 2012" to be different tasks. This allows Gilad to

1.4. **Allocation.**

1.5. **Objective function.** Each crop $c \in C$ has a yield of $y_{c,w}$, depending on the week $w \in W$ it is planted. The objective function we want to maximize 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] C. Artigues, O. Koné, P. Lopez, and M. Mongeau, *Comparison of mixed integer linear programming models for the resource-constrained project scheduling problem with consumption and production of resources*. http://www.math.univ-toulouse.fr/~mongeau/MILP-RCPSP-CPR-submitted.pdf, February 2011.

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

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

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