Solving The Automated Vacuum Waste Collection Optimization Problem ⋆

Ramón Béjar[1], Cèsar Fernández[1], Carles Mateu[1], Felip Manyà[2], and Francina Sole-Mauri[1]

[1] Dpt. of Computer Science, Universitat de Lleida, Spain
[2] IIIA-CSIC, Spain

Introduction

Automated vacuum waste collection (AVWC) uses air suction on a closed network of underground pipes to transport waste from the drop off points scattered throughout the city to a central collection point, reducing greenhouse gas emissions and the inconveniences of conventional methods (odors, noise, ...). Since a significant part of the cost of operating AVWC systems is energy consumption, we have started a project with the aim of applying constraint programming technology to schedule the daily emptying sequences of the drop off points in such a way that energy consumption is minimized.

In this paper we define AVWC systems, encode the problem of deciding the drop off points that should be emptied at a given time as a constraint integer programming (CIP) problem, and empirically evaluate our approach with real data.

Our paper is a step forward in solving the challenge posed in [1], and extends the results of [2] by dealing with a limited but complete set of sectors, and by making decisions taking into account subsectors instead of individual drop off points. This way, we obtain simpler models that achieve good quality real-time solutions. Furthermore, we include experiments with real data that were beyond the reach of [2].

AVWC systems and its optimization problem

An AVWC system is defined by a set $\{\mathcal{T}, \mathcal{I}, \mathcal{F}, \mathcal{V}^a, \mathcal{V}^s\}$, where $\mathcal{T}(\mathcal{N}, \mathcal{E})$ is a rooted binary tree with nodes ($\mathcal{N}$) representing either inlets ($\mathcal{I}$) or pipe junctions, and edges ($\mathcal{E}$) correspond to union pipes between nodes; $\mathcal{F}$ is the set of waste fractions; $\mathcal{V}^a$ is the set of air valves, located at some inlets for creating air streams able to empty downstream inlets; and $\mathcal{V}^s$ is the set of sector valves that segment the whole tree defining isolated sectors (s). Sectors are subtrees of $\mathcal{T}$, always containing the root node and a subset of $\mathcal{I}$ ($\mathcal{I}^s$). Each inlet in $\mathcal{I}$ is denoted by I_i^f (f is the fraction), meanwhile v_i^a and v_i^s denote air and sector valves, respectively (see Fig. 1).

Each inlet is associated with three subtrees: (i) the emptying subtree ($\mathcal{T}_i^E$) is the path that waste must follow from inlet i to the root node (central collection point);

⋆ Research partially funded by projects: ARINF (TIN2009-14704-C03-01/03), TASSAT (TIN2010-20967-C04-01/03) and Newmatica (INNPACTO IPT-2011-1496-310000). Projects funded by MICINN (until 2011) and MINECO (from 2011).

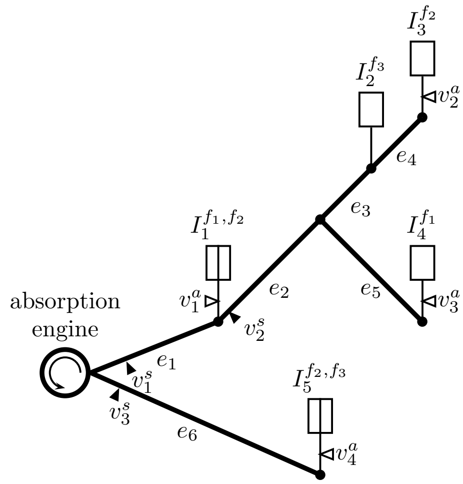

Fig. 1. Schematic example of an automatic vacuum waste collection plant

(ii) the air subtree ($\mathcal{T}_i^A$), where $\mathcal{T}_i^E \subseteq \mathcal{T}_i^A$, being equal if inlet i has an air valve; and (iii) the vacuum subtree ($\mathcal{T}_s^V$) that represents the total amount of air to be moved before proceeding to waste transport.

Let L_i^f be the waste occupancy of an inlet I_i^f at the beginning of slot time t. Given a sector s and fraction f, a valid emptying sequence $\mathcal{E}_t^{f,s} = [I_{i_1}^f, I_{i_2}^f, \ldots]$ is an ordered subset of the inlets in sector s such that the total waste to be emptied does not exceed a maximum transfer capacity (L_{max}^f): $\sum_{I_i^f \in \mathcal{E}_t^{f,s}} L_i^f \leq L_{max}^f$.

The energy consumption of an emptying sequence depends on the air speed operation (v_t), which is constant for each sequence, and the operation time (T_t), which is the time required to operate an emptying sequence. The operation time is divided into two phases. In the transitory phase the previous speed (v_{t-1}) changes progressively to v_t in time T_t^{tr}, and in the stationary phase the selected emptying sequence is executed in time T_t^{st}.

Energy can also be split into two parts: transitory (E_t^{tr}) and stationary (E_t^{st}). In the transitory part, there is only energy consumption for the process of increasing air speed. In the stationary part, for the same emptying path, the minimum transport energy is obtained when the shortest air path is employed, that is, opening the upstream air valve closest to the inlet being emptied. The type of fraction also affects the power requirements, needing more energy those types of fraction more dense. Under these considerations, the contribution to the stationary energy of an inlet I_i^f of the emptying sequence is proportional to its air path ($\mathcal{T}_i^A$) and to its transport time up to the next intersection.

The AVWC optimization problem consists in finding a set of emptying sequences and air speed operations, $\{\mathcal{E}_t^{f,s}\} \times \{v_t\}$, $0 \leq t \leq T$, for a period of time T (e.g. a day), that minimizes the energy cost: $\sum_{t=0}^{T} f_c(t) \cdot (E_t^{tr} + E_t^{st})$, where $f_c(t)$ depends on time and energy fares. Also, at the end of the period T, the residual load L_i^f of inlet I_i^f should be below a lower bound ϵ_i^f.

In this paper we only deal with the problem of selecting an optimal emptying sequence on a given time slot t, leaving as future work the problem of optimizing over a full period of time T, i.e. the dynamic continuous problem. The problem at time t will be subject to the following hard constraints and conditions:

- The emptying sequence $\mathcal{E}_t^{f,s}$is valid: $\sum_{I_i^f \in \mathcal{E}_t^{f,s}} L_i^f \leq L_{max}^f$.
- The air speed is between a minimum (V_i^f) for each inlet and a maximum (V_M): $\max_{I_i^f \in \mathcal{I}^s}(V_i^f) \leq v_t \leq V_M$.
- Any inlet I_i^f with a load L_i^f over a threshold th_i^f should be included in the emptying sequence. Given that it may not be possible to include all the inlets that are overloaded, we force the inclusion of the maximum number of such inlets by a penalty term in our objective function.

The CIP encoding

Our CIP encoding is a variant of the encoding in [2], which is simpler but produces good quality real-time solutions in large AVWC systems. The main differences are:

- Both encodings do not consider all the possible valid sectors that can be obtained by closing and opening the valves in $\mathcal{V}^s$. They just select a subset of those sectors, but now we introduce the notion of subsector. A subsector is a subset of inlets belonging to a sector, and from the encoding point of view, subsectors are encoded as sectors. As all the subsectors of a given sector have the same $\mathcal{V}^s$ configuration, all of them present the same $\mathcal{T}_s^V$ as the sector s which they belong to.
- The new encoding only considers the inlets with a load above a threshold (40% in our experiments).
- The ability to define disjoint and small subsectors reduces the number of variables. The worst-case number of variables of our encoding depends quadratically on the number of inlets per sector.

Experimental Results

As an example, we applied our CIP encoding to a real AVWC system with 5 sectors, 36 drop off points, 124 inlets and 4 fractions. We used real drop off data of the most and the less loaded day in the season, representing a transport of 39.4 m^3 and 31.8 m^3, respectively. Each daily data set has 288 inputs corresponding to an inlet volume load sampling rate of 5 minutes. The encoding was solved with SCIP version 2.1.1[3] with SoPlex 1.6 and default settings, in a 2.66 GHz processor. The timeout was set to 5 minutes, and the maximum load of the system to 1 m^3.

Figure 2 is a histogram of the time needed to solve the instances. Each one of the daily data (high and low load) is solved, first, according to the sectors defined for the

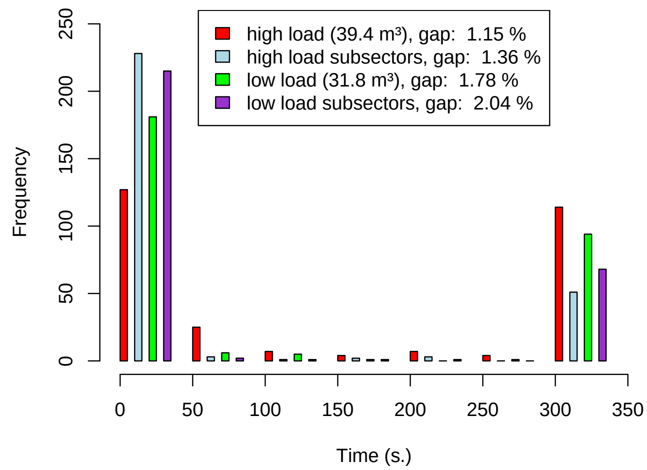

Fig. 2. Histogram of time to solve on a real AVWC system

topology, and second, adding some subsectors. Actually, only two subsectors are defined on two different sectors. As shown, most of the instances are optimized in a few seconds, being the subsectorized problems even more efficient. Even for those instances reaching the time out, the solver is always able to find a good near optimal solution, as the relative difference between the primal and dual bounds (mean gap value) indicates. Such a good encoding performance indicates that our solving approach could be a fundamental part of a planning algorithm, maybe based on dynamical programming, that works over a daily time horizon, allowing an efficient learning based on historical inlet disposal data, as well as an optimal, or near-optimal, real time decision algorithm.

References

1. Béjar, R., Fernández, C., Manyà, F., Mateu, C., Solé, F., Vidal, D.: The automated vacuum waste collection optimization problem. In: Proceedings of the AAAI 2012 (to be published), AAAI Press / The MIT Press (2012)
2. Béjar, R., Fernández, C., Manyà, F., Mateu, C., Solé, F.: Optimizing energy consumption in automated vacuum waste collection systems. In: Proceedings of the CP 2012 (submitted), Springer (2012)
3. Berline, Z.I.: SCIP, solving contraint integer programs. `http://scip.zib.de` (2012)