Proceedings of the Federated Conference on Computer Science and Information Systems pp. 303–306 DOI: 10.15439/2020F13 ISSN 2300-5963 ACSIS, Vol. 21

Pipe-lining Dynamic Programming Processes in Order to Synchronize Energy Production and Consumption

Fariha Bendali
LIMOS CNRS/UCA
Clermont-Fd, France
Email: bendali@isima.fr

Alain Quilliot
LIMOS CNRS UMR 6158
LABEX IMOBS3, Université Clermont-Auvergne
Bat. ISIMA, BP 10125
Campus des Cézaux,
63173 Aubière, France
Email: quilliot@isima.fr

Eloise Mole Kamga
LIMOS CNRS UMR 6158
63173 Aubière, France
Email: eloise@isima.fr

Jean Mailfert
LIMOS CNRS/UCA
Clermont-Fd, France
Email: mailfert@isima.fr

Hélène Toussaint
LIMOS CNRS UMR 6158
LABEX IMOB3
Clermont-Fd, France

***Abstract*—Synchronizing heterogeneous processes remains a difficult issue in Scheduling area. Related ILP models are in trouble. So we propose here a pipe-line collaboration of a dynamic programming process for energy production and consumption scheduling.**

I. Introduction

EFFICIENTLY synchronizing heterogeneous process remains a difficult issue when it comes to scheduling. ILP models are flawed by large gaps induced by the relaxation of the integrality constraint (the *Big M* problem). This difficulty also arises when one wants to plan industrial, domestic or local logistics activities, while relying on local renewable energy production: Due to both market deregulation and emergent technologies, the rise local producers (factories, farms,…) while simultaneously remaining consumers (see [1, 6]) tends to make this issue a trend in Energy Economics. In the context of Labex IMOBS3 project in Clermont-Fd, France, devoted to *Innovative Mobility*, we are involved into the control of a local micro-plant for hydrogen (H^2) production, which provides autonomous vehicles in H^2 fuel. Researchers rely here on solar power and photolysis ([4, 6, 7]): so the productivity of the process deeply depend on the intensity of solar illumination. Few works address resulting synchronization issue (see general contribution in [1], studies about electric vehicles (*Green VRP*, *Pollution-Routing Problem*,…in [3, 8]), and industrial processes (see [2, 5]) under time-dependent energy costs and access restrictions.

Because of the IMOBS3 project, present contribution is about the synchronous management of, on one side, a fleet of small electric vehicles provided with H^2 power cells, and, on the other side, a micro-plant in charge of local H^2 fuel production. Taken as a whole, resulting model of Section II involves forecasting, safety management and scheduling. We only address here the last issue, while considering only one vehicle, required to perform tasks according to a pre-fixed order, which periodically goes back to the micro-plant in order to refuel. The micro-plant has its own production/storage restrictions. Relying on ILP is inefficient, and so we first propose in Section III an exact *Dynamic Programming Scheme* (DPS). But, though this DPS allows us to state a PTAS (*Polynomial Time Approximation Scheme*) result, it remains time-costly in practice. So we decompose it (Section IV) into 2 DPS sub-processes, one related to the vehicle, and the other one to the micro-plant which collaborate through a *pipe-line*.

II. The Energy Production/consumption problem (EPC)

Some vehicle has to perform internal logistics tasks, while following a route Γ which starts from some *Depot* node and ends in the same way after going through stations $j = 1, \dots, M$, according to this order. Start-node *Depot* has label 0 and End-node *Depot* has label $M + 1$. The time required by the vehicle in order to go from j to $j + 1$ is equal to t_j, (including *service* time). The vehicle may leave *Depot* at time 0 and should finish its route no later than some threshold time *TMax*. It is powered by hydrogen (H^2) fuel. The capacity of its tank is denoted by C^{Veh} and we know, for any $j = 0, \dots, M$, the H^2 amount e_j required in order to move from station j to station $j + 1$. The initial H^2 load of the vehicle is denoted by E_0, and the vehicle is required to end its trip with at least the same energy load. It comes that the vehicle must periodically refuel. Refueling transactions take place at a *micro-plant*, close to *Depot*: The time required by the vehicle in order to move from station j to the micro-plant (from the micro-plant to j) is denoted by d_j (d^*_j); by the same way, the energy required in order to move from j to the micro-plant (from the micro-plant to j) is denoted by ε_j (ε^*_j). Figure 1 displays an example of a trip performed by the vehicle along station *Depot* = 0, 1, 2, 3, 4, 5, 6 = *Depot*.

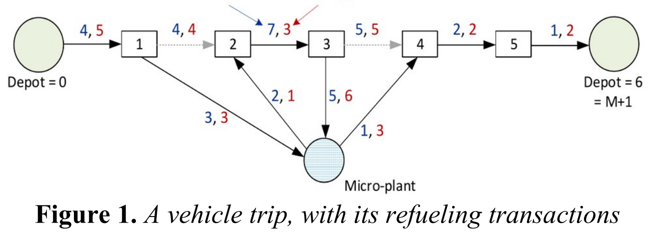

Figure 1. *A vehicle trip, with its refueling transactions*

On another side, the micro-plant produces H^2 *in situ* through photolysis&electrolysis. Resulting H^2 is stored inside the micro-plant's tank, with capacity equal to C^{MP}. We suppose that the time space {0, ..., $TMax$} is divided into periods $P_i = [p.i, p.(i + 1)[$, $i = 0, \ldots, N - 1$, with $TMax = N.p$ (see Figure 3). We identify index i and period P_i. If the micro-plant is *active* at some time during period i, then it is active during the whole period i, and produces R_i hydrogen fuel units. At time 0, the load of the micro-plant tank is $H_0 \leq C^{MP}$ and the micro-plant is idle. This should also hold at time $TMax$. Because of safety concerns, the vehicle cannot refuel while the micro-plant is producing and any vehicle refueling transaction takes a whole period i. Besides, producing H^2 fuel has a cost, which may be decomposed into:

- A constant *activation cost* $Cost^F$, which is charged every time the micro-plant is activated.

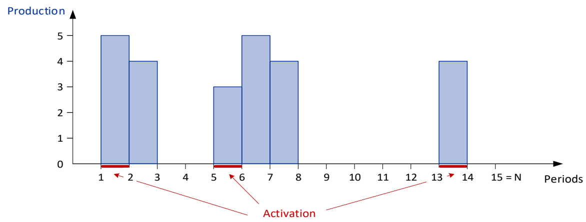

Figure 2. ***An example of micro-plant activity, with N = 15***

- A *time-dependent* production cost $Cost^V_i$ which reflects the time-indexed prices charged by the electricity provider.

Then the *Energy Production/Consumption* (**EPC**) Problem consists in scheduling both the vehicle and the micro-plant in such a way that:

- The vehicle starts from *Depot* = 0, visits all stations $j = 1, \ldots, M$ and comes back to *Depot* at some time $T \in [0, TMax]$, while refueling every time it is necessary;
- The micro-plant produces and stores in time the H^2 fuel needed by the vehicle;
- Both induced H^2 production cost *Cost* and time T are the smallest possible: $Min = Cost + \alpha.T$, where α is some scaling coefficient.

Figure 3 below shows the synchronization between the vehicle and the micro-plant of fig. 1, 2, in case $p = 2$, $E_0 = 8$, $H_0 = 4$, $TMax = 30$, $Cost^F = 7$, $C^{MP} = 15$, $C^{Veh} = 15$, $\alpha = 1$.

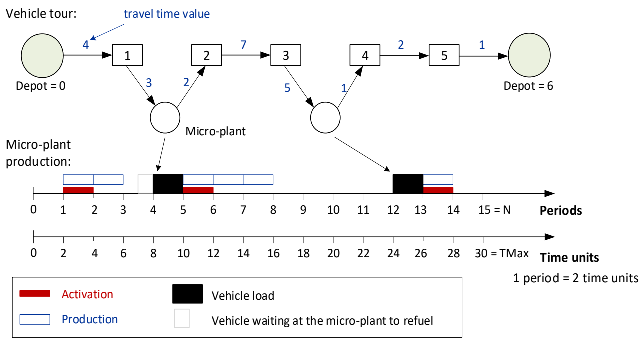

Figure 3. *A feasible solution related to Fig. 1 and 2.*

III. A *DPS-EPC* ALGORITHM.

EPC is *NP-Hard*: It can be reduced to Knapsack. We first handle it through DPS (*Dynamic Program Scheme*):

DPS *Time Space* and *States*: The *time space* is the set Δ of *time pairs* (i, j), $i = 0, \ldots, N$, $j = 0, \ldots, M + 1$.We link periods i and stations j through relations (<<, >>, ==) which locate period i with respect to time value $T \in \{0, \ldots, TMax\}$:

- $T << i$ if $T < p.i$; $T >> i$ if $T \geq p.(i + 1)$;
- $T == i$ if $p.i \leq T < p.(i + 1)$.

For any such a *time pair* (i, j), a related *state* is a 4-uple $s = (Z, T, V^{Tank}, V^{Veh})$, with:

- $Z = 1$: micro-plant active at the end of period $i - 1$.
- V^{Tank} and V^{Veh} are respectively the loads of the micro-plant at the beginning of i and the vehicle when it arrives at j;
- T is a value in 0, …, $TMax$ with the meaning:
  - $T >> i$: the vehicle will reach j at time T;
  - $T << i$: the vehicle is between j and the micro-plant, possibly waiting for being refueled;
  - $T == i$: the vehicle is in j, and decides between riding to $j + 1$ or to the micro-plant.

Initial *state* corresponds to *time pair* (0, 0) and 4-uple $s_0 = (0, 0, H_0, E_0)$. Final *state* corresponds to any *time pair* $(i \leq N, M + 1)$, and any 4-uple $(Z, T \leq TMax, V^{Tank} \geq H_0, V^{Veh} \geq E_0)$.

Decisions/Transitions/Costs. Then a decision D is a 3-uple $D = (z, x, \delta)$ in $\{0, 1\}^3$, with the meaning:

- $z = 1$ ~ the micro-plant produces during period i;
- x refers the case $T == i$: $x = 0$ means that the vehicle rides from j to $j + 1$ without refueling; $x = 1$ means that it refuels at the micro-plant while ridingfrom j to $j + 1$.
- $\delta = 1$ ~ the vehicle is located at the micro-plant and decides to refuel during period i, forbidding the micro-plant to be active during this period. It requires $T << i$ and $p.i - T \geq d_j$.

Decision is taken at the end of period $i - 1$. For any *time pair* (i, j) and *state* $s = (Z, T, V^{Tank}, V^{Veh})$, no more than 4 decisions D are feasible:

- **1 th case**: $T >> i$. Then the only choice is about z.
- **2 th case:** $T << i$ and $p.i - T < d_j$. The vehicle is moving from j to the micro-plant and cannot refuel yet. Once again, the only choice is about z.
- **3 th case:** $T << i$ and $p.i - T \geq d_j$. Then, we have 3 choices: 1). **Producing**: $z = 1$; $\delta = 0$; 2). **Refueling**: $z = 0$; $\delta = 1$; 3). **Doing nothing**: $z = 0$; $\delta = 0$. l.
- **4 th case:** $T == i$. Then we have 4 choices:
  - **Producing and riding towards *j*+1:** $z = 1$, $x = 0$.
  - **Not Producing, riding towards *j*+1:** $z = 1$, $x = 0$.
  - **Not Producing, riding to micro-plant:** $z = 0$, $x = 1$.
  - **Producing, riding to micro-plant:** $z = 1$, $x = 1$.

We implement Bellman Equations through a *Forward Driven* Strategy and denote by ***DPS-EPC*** the algorithm designed this way. In order to control th number of states, we need to enhance it with filtering devices.

A. Filtering through Rounding: A PTAS Result.

DPS-EPC is in trouble when M and N are large. Still, by considering that 2 states are equivalent when they are equal modulo the K largest bits and extending the notion of state in a well-fitted way, we turn ***DPS-EPC*** into an algorithm ***DPS-EPC***(K) which allows to state:

Theorem 2 (Polynomial Time Approximation Scheme): *For any value $\varepsilon > 0$, we may choose $K = K(\varepsilon)$ large enough in such a way that in case EPC admits an optimal solution with value W^{Opt}, then **DPS-EPC($K(\varepsilon)$)** yields in polynomial time a solution which is feasible with regards to initial values $(1 + \varepsilon/2).H_0$ and $(1 + \varepsilon/2).E_0$, threshold values $(1 + \varepsilon).C^{MP}$, $(1 + \varepsilon).C^{Veh}$ and $(1 + \varepsilon).TMax$ and whose cost value is no larger than W^{Opt}.*

B. Logical Filtering Devices.

First, we apply the standard *Dominance* Rule: If, for a given time pair (i, j), state s_1 dominates state s_2, ($W_1 \leq W_2$; $T_1 \leq T_2$; $Z_1 \geq Z_2$; $V^{Tank}{}_1 \geq V^{Tank}{}_2$; $V^{Veh}{}_1 \geq V^{Veh}{}_2$), then we *kill* s_2. But this has little filtering power. So, for any time pair (i, j), and related state $s = (Z, T, V^{Tank}, V^{Veh})$, we get rough estimations *Fuel* and *Time* of respectively energy and time required in order to allow the vehicle to return from j to *Depot*, and derive the following *logical filtering rules*:

1). ***Makespan Based*** **filtering rule**: If ($Time \geq TMax - T + 1$) then *kill state* $s = (Z, T, V^{Tank}, V^{Veh})$ related to *time pair* (i, j), since there is not enough time left for the vehicle to achieve its trip.

2) ***Energy Based*** **filtering rule**: If $Fuel > V^{Veh} \Sigma_{k \geq i} R_k + V^{Tank}$ then *kill state* $s = (Z, T, V^{Tank}, V^{Veh})$ related to *time pair* (i, j), since there won't be enough energy for the vehicle to achieve its trip.

We go further and pre-compute, for any energy amount V, any period number i, and any micro-plant Z value, the minimal cost $Cost\text{-}Min(i, V, Z)$ required from the micro-plant to produce V energy units from time $p.i$ on, Z denoting the state of the micro-plant at the end of period $i - 1$. Then, for any *time pair* (i, j) and any state $s = (Z, T, V^{Tank}, V^{Veh})$ with value W, we derive a lower bound LB of a best **EPC** trajectory involving (i, j) and s, by setting: $LB((i, j), s) = \alpha.Time + Cost\text{-}Min(i, (Fuel - V^{Tank})^+, Z) + W$. This lower bound allows us to turn ***DPS-EPC*** into a greedy procedure ***GREEDY-EPC***, by keeping, for any time pair (i, j), only the state $s(i,j)$ which minimizes $LB((i, j), s)$. ***GREEDY-EPC*** provides us with some feasible value *Current-Value* and we may apply the following ***Upper/Lower Bound Based*** **filtering rule** 3): If $LB((i, j), s) \geq Current\text{-}Value$, then *kill state* $s = (Z, T, V^{Tank}, V^{Veh})$, related to time pair (i, j).

IV. PIPE-LINE DECOMPOSITION OF *DPS-EPC*.

*A. The **DPS-Vehicle** Scheme.*

We do here as if micro-plant were able to provide, at any time, the vehicle with as much as energy it needs. we optimize the *Refueling Strategy* of the vehicle, that is the {0, 1} valued vector $x = (x_j, j = 0..M)$ and the load vector $L = (L_j, j = 0..M)$ which tell us at which stations j vehicle will refuel between j and $j+1$, and how much, while minimizing some quantity: $\alpha.T^{End} + \beta.(\Sigma_j L_j.x_j)$, where T^{End} means the ending date of the vehicle trip, and β is an auxiliary *cost* coefficients. We notice that every time the vehicle arrives to the micro-plant, it is sufficient for him to refuel exactly the H^2 it needs in order to reach the next refueling transaction. This leads us to the following ***DPS-Vehicle*** scheme, whose components *time*, *state* and *decision* come as follows:

- **Time Space**: the set $J = \{0, 1, \ldots, M, M+1\}$.
- **State Space**: A state s is a 2-uple $s = (T, V^{Veh})$: T is the time necessary in order to come back to *Depot* and V^{Veh} the load at j of the vehicle tank. Its *value* $W = \alpha.T + \beta.U$, involves the energy amount U which will be wasted by the vehicle before the end of its trip. **Initial state** (in the sense of a ***backward driven*** DPS) is the state $(0, E_0)$ related to $j = (M+1)$. **Final states**, related to $j = 0$, should be any state ($T \leq TMax$, $V^{Veh} \leq E_0$).
- **Decision Space**: A decision $x \in \{0, 1\}$: $x = 0$ means a no *refueling move* to $j+1$, and $x = 1$ means a *refueling move* to the micro-plant before reaching $j+1$.
- **Backward Driven Strategy**: In order to store, for any pair (j, V^{Veh}), the time and energy amount required in order to achieve tour, we implement Bellman Principle according to a backward driven strategy.

We denote by ***DPS-Vehicle*** the resulting DPS algorithm. In order to synchronize it with the H^2 Production control, we retrieve from any run a *Reduced Refueling Strategy*, that is:

- S = number of refueling transactions; Loads μ_s = quantities of H^2 which is loaded for every value $s = 1\ldots S$;
- Lower bounds $m_1,..m_Q$ and upper bounds $M_1,..,M_S$ for the related period numbers $i_1,.., i_S \in \{0,.., N\text{-}1\}$, as well as *Time Lag* coefficients $B_1, .., B_S$ which means: For any $s = 1..S\text{-}1$, $i_{s+1} \geq i_s + B_s$.

*B. The **DPS-Prod** Scheme*

Let S, m, M, μ be a *Reduced Refueling Strategy*, as above. Then we want to schedule the activity of the micro-plant, that is compute {0,1}-valued vectors z and δ with indexation on $i = 0..N\text{-}1$ as in ***DPS-ECP***, in such a way that:

- The vehicle may refuel at some periods $i_1,.., i_S$ in a way consistent with time lags and time window constraints induced by the *Reduced Refueling Strategy*
- The micro-plant ends with the H^2 load as it started;
- We minimize $\alpha.i_S + \Sigma_{i = 0..N\text{-}1} (Cost^F.y_i + Cost^V{}_i.z_i)$.

We apply a **forward driven** DPS algorithm ***DPS-Prod*** with the following *Time, State,* and *Decision* components:

- **Time Space**: the set $I = \{0..N\}$.
- **State Space**: For any $i = 0..N$, a state is a 4-uple $E = (Z, V^{Tank}, Rank, Gap)$, with *Rank* in $0..S$:
  - $Z = 1$ ~ the micro-plant is active at the end of i-1.
  - V^{Tank} is the load of the micro-plant when i starts.
  - $Rank \in 1..S$ ~ the $Rank^{th}$ refueling transaction has been performed and we are waiting for the $(Rank + 1)^{th}$ refueling transaction. *Gap* means the difference between i and the period when the $Rank^{th}$ refueling transaction was performed.

  For every $i = 0..N$, a state E is provided with its current Bellman value W^{Prod}.
  - **Initial state** is $E^{Start} = (0, H_0, 0, 0)$, with related value $W^{Prod} = 0$, and time value $i = 0$;
  - **Final states** are states $E^{End} = (Z, V^{Tank} \geq H_0, S, 0)$, associated with a time value $i \leq N$;
- **Decision/Transitions**: For any $i = 0..N$, $E = (Z, V^{Tank}, Rank, Gap)$, a decision is defined as a 2-uple (z, δ) in $\{0,1\}^2$, with the following meaning:
  - $z = 1$ ~ the micro-plant will produce during period i;
  - $\delta = 1$ ~ the vehicle will perform its $(Rank+1)^{th}$ refueling transaction during period i.

  Since production and refueling cannot be performed simultaneously, there are only 3 possible decisions:
  1). $\underline{z = 1, \delta = 0}$; 2). $\underline{z = 0, \delta = 0}$; 3). $\underline{z = 0, \delta = 1}$.

As for ***DPS-EPC***, we may enhance ***DPS-Prod*** through *logical* and *upper/lower bound based* filtering devices.

C. The Pipe-Line Scheme

Clearly, the simplest way to make above ***DPS-Vehicle*** and ***DPS-Prod*** interact, is to design the following heuristic ***Pipe-Vehicle->Production***:

Main Steps of *Pipe-Vehicle->Production*:
1). Fix β and Apply ***DPS-Vehicle***, to the *Vehicle* instance related to α, β: get related *Reduced Refueling Strategy*;
2). Apply ***DPS-Prod*** to resulting *Production* instance;
3). **Reconstruct the whole EPC solution**.

Choosing β: β should reflect the energy production cost. Since we do not know when the refueling transactions take place, we do as if were to be uniformly distributed.

V. NUMERICAL EXPERIMENTS.

Purpose and Technical Context: We evaluate: 1). the pipe-line decomposition ***DPS-Vehicle*** and ***DPS-Prod***; 2). the filtering devices and the greedy procedure described in III.2, while using C++, on Windows 10 with IntelCore i5-6500@3.20 GHz CPU, 16 Go RAM.

Instances: We fix N and M, and randomly generate stations j and *Depot* and the *Micro-Plant* as point of the R^2 space. Then d_j, d^*_j and t_j, e_j, ε_j, ε^*_j respectively corresponds to Euclidean and Manhattan distances. Then we fix C^{MP}, C^{Veh}, $TMAX$, $Cost^F \geq \text{Inf}_i\, Cost^V_i$, $i = 0,\ldots, N-1$.

Outputs: We first run ***Greedy-EPC*** with 50 replications, => gap *G-Gap* to optimality. Next we run ***DPS-EPC***:
1) Only with the *Strong Dominance Rule* => $ST(1)$ = Maximal number of states for a given pair (i,j),
2) With the *2 Logical Filtering* rules => $ST(2)$;
3) With all filtering rules => $ST(3)$ (i,j).

Next we run ***Pipe-Vehicle->Production*** and get max state/time *ST-Veh*, *ST-Prod*, and gap *P-Gap* to optimality.

Instance (*M*, *N*)	*G-GAP*	*ST*(3)	*ST*(2)	*ST*(1)
1, (6, 27)	18.4	11870	11369	394754
2, (6, 26)	1.5	447	2619	299933
3, (10, 25)	0.0	10642	25636	107228
4, (10, 31)	11.4	17526	26254	310543
5, (10, 46)	0.0	21404	45014	425009

TABLE 1: VALUES *N, M, G-Gap, ST(1), ST(2), ST(3)*

Inst (*M, N*)	*ST-Veh*	*ST-Prod*	*P-Gap*
1, (6, 27)	22	895	6.4
2, (6, 26)	18	105	0
3, (10, 25)	43	902	0
4, (10, 31)	50	1088	2.9
5, (10, 46)	52	1385	0

TABLE 2: VALUES *N, M, ST-Veh, ST-Prod, P-Gap*

Comment: *Dominance* rule has little impact, logical anticipation and optimistic estimation rules are significantly more efficient, while the pipe-line scheme ***DPS-Vehicle* -> *DPS-Production*** offers a good tradeoff time/accuracy.

VI. CONCLUSION

In the future, we shall deal with uncertainties, address the *vehicle route* issue, manage the *on line* context.

REFERENCES

[1]. S.Albers: *Energy-efficient algorithms*; **Communications of ACM** 53, 4, p 86-96, (2010). https://dl.acm.org/doi/10.1145/1735223.1735245

[2]. ARTIGUES, E.HEBRARD, A.QUILLIOT, H.TOUSSAINT: *Models and algorithms for evacuation problems*; **IEEE Proc. FEDCIS WCO12**, Leipzig, 4 pages, (2019). https://doi.org/10.15439/2019F90

[3]. L.Benini, , A.Bogliolo, G.De Micheli: *A survey of design techniques for system level dynamic power management*; **IEEE Transactions of Very Large Scale Integratio Systems**, 8, 3, p 299-316, (2000). https://dl.acm.org/doi/10.1145/1403375.1403402

[4]. C.C.Chan. *The state of the art of fuel cell vehicles*. **Proc. of the IEEE**, 95, p 704-718, (2007). DOI: 10.1109/JPROC.2007.892489

[5]. P.Chretienne, A.Quilliot: *A polynomial algorithm for the non idling scheduling problem*; **DAM**, 20 pages, (2018). https://doi.org/10.1016/j.dam.2013.01.019

[6]. C.Grimes, O.Varghese, S.Ranjan. *Light, water, hydrogen: photoelectrolysis*. **Springer US**, (2008). ISBN 978-0-387-33198-0

[7]. S.Licht. *Thermochemical and Thermal/Photo Hybrid Solar Water Splitting*, **Springer New York, NY**, (2008). https://link.springer.com/chapter/10.1007/978-0-387-72810-0_5

[8]. C. Lin, K.L.Choy, G.T.Ho, S.H. Chung, H.Lam. *Survey of green vehicle routing problem*. **Expert Systems Applications**, 41, p 1118–1138, (2014). https://doi.org/10.1016/j.eswa.2013.07.107.