# A constraint-programming based decomposition method for the Generalised Workforce Scheduling and Routing Problem (GWSRP)

Eric Bourreau, Thierry Garaix, Matthieu Gondran, Philippe Lacomme, Nikolay Tchernev

HAL Id: lirmm-03286971

https://hal-lirmm.ccsd.cnrs.fr/lirmm-03286971

Submitted on 15 Jul 2021

# A constraint-programming based decomposition method for the Generalised Workforce Scheduling and Routing Problem (GWSRP)

E. Bourreau[a], T. Garaix[b], M. Gondran[c], P. Lacomme[c,1], N. Tchernev[c]

[a] LIRMM, UMR 5506, 161 rue Ada, 34392 Montpellier Cedex 5, France.

[b] École des Mines de Saint-Étienne, 158 cours Fauriel, 42000 Saint-Étienne, France.

[c] Laboratoire d'Informatique (LIMOS, UMR CNRS 6158), Campus des Cézeaux, 63177 Aubière Cedex, France.

Recent studies prove that industry is confronted with workforce issues which are, the majority of the time, determining factors in its evolution over the long-term. The aims of workforce management are to balance numerous objectives including relations between jobs/workers (workers' skills), operational costs, customers' quality of service and workers' quality of service. In their workforce management, many manufacturers include the routing of workers between customers. Another challenging and important difficulty is the coordination between workers to perform a job or a service. This paper deals with the Generalised Workforce Scheduling and Routing Problem (GWSRP) where 9 temporal constraints ensuring visit dependencies are all together taken into account and where customers and workers' quality of service are taken into consideration. A Constraint-Programming based Decomposition Method (CPDM) is proposed, firstly based on a relaxation of coordination constraints, and secondly with a constraint programming approach taking coordination constraints into account. Numerical experiments are achieved on instances derived from WSRP benchmark instances with up to 177 customers and 59 vehicles.

**Keywords***: Workforce Scheduling and Routing Problem, Dynamic programming, Constraint Programming, Column generation, Temporal dependencies

## 1 Introduction and literature review

Workforce scheduling is a crucial point for manufactures in order to be highly competitive in modern industrial organisations (Mundschenk and Drexl, 2007), (Saadat et al., 2013). An effective management of manpower resources and skills is one of the most critical functions affecting performance in many industries (Wirojanagud et al., 2007), (Attia et al., 2014), in particular, concerning maintenance activities (Wongmongkolrit and Rassameethes, 2010).

A vast majority of the publications assume that workers are identical (Wirojanagud et al., 2007), nevertheless, individual worker differences may affect workforce planning and management decisions at a strategic level and therefore individual worker differences are taking into consideration in recent articles (Maenhout and Vanhoucke, 2009), (Wongmongkolrit and Rassameethes, 2010), (Attia et al., 2014). For example, workers are cross-trained, have different costs, different working hours preferences or different customer preferences.

Workforce Problem With Scheduling and Routing (WPWSR) refers to a problem in which workers must carry out jobs at different locations hence be required to take transport into consideration (Castillo-Salazar et al., 2016). WPWSR are an important challenge for companies involved in two main sectors: production and/or maintenance firms, and home health care services. (Xu et al., 2015) assert that the main challenge of WPWSR is to assign the right worker

---

[1] Corresponding author. e-mail addresses: placomme@isima.fr (P. Lacomme),

to customers and plan an appropriate trip for each worker with real-world constraints. This assignment must be a compromise between the operational costs, the customer satisfaction and the employees' working conditions. (Xu et al., 2015) point out that: on the one hand, the majority of literature focuses on operational cost optimisation without any consideration for customers and workers satisfactions; and on the other hand, the coordination (or cooperation) of the workers has not received much attentions.

Production and/or Maintenance Problems (PMP) (Koochaki et al., 2013) and Home Health Care (HHC) problems (Bachouch et al., 2011), (Suter et al., 2014) are similar in that: workforce scheduling and routing are crucial; processing (or service) times and transportation times are taken into account; and workers can be different. These problems are equivalent to vehicle routing problems with some specific constraints as highlighted by (Mathlouthi et al., 2018) for PMP and by (Bachouch et al., 2011) for HHC problem.

WPWSR are characterised by many objectives complying with different priority levels, which model the customers' quality of service, the operational costs, and the workers' preferences. PMP and HHC problems tend to incorporate cooperation and coordination between workers (Chankov et al., 2018), (Bachouch et al., 2011).

| Nomenclature | | | |
|---|---|---|---|
| $W$ | set of workers | $\rho_j^w$ | Quality of Service (QoS): $\rho_j^w \in [0; 3]$ |
| $A$ | set of visits | $PA_j^w$ | preference for $w$ to perform $j$, $PA_j^w \in \{0; 1\}$ |
| $T_{i,j}$ | time to travel from visit $i$ to visit $j$ | $[E_j; L_j]$ | time window of visit $j$ |
| $PrC_j^w$ | worker $w$'s cost to perform visit $j$ | $[TW_{inf}^w; TW_{sup}^w]$ | working time window for worker $w$ |
| $Pt_j$ | processing time of visit $j$ | $\{E_{C1}, E_{C1}, ..., E_{C9}\}$ | The 9 sets of coordination constraints |
| $I_w$ | initial depot node of worker $w$ | $\theta_j^w$ | $w$'s time window violation when performing $j$ |
| $L_w$ | final depot node of worker $w$ | $\psi_j^w$ | $w$'s area violation when performing $j$ |
| $Com_j^w$ | =1 if worker $w$ is allowed to perform visit $j$, 0 otherwise | | |

This paper focuses on a new integrated problem, the Generalised Workforce Routing and Scheduling Problem (GWSRP), which combines an assignment problem, a routing problem, and a scheduling problem with: temporal dependencies (also referred to as coordination constraints), non-identical workers, customers' quality of service; and workers' quality of service. The GWSRP is an extension of the WSRP (Castillo-Salazar et al., 2016) that provides an efficient modelling of numerous routing and scheduling problems.

This paper presents:

- A Constraint-Programming based Decomposition Method (CPDM) in three steps: the first one creates trips by a column generation, without any temporal constraint consideration, and where the slave problem is an extension of the Elementary Shortest Path Problem with Resource Constraints (ESPPRC). The second step is a heuristic to solve the set partitioning problem maximising the number of coordination constraints satisfied. The third step is a trip reconstruction which enable the investigation of trips in order to include all of the remaining temporal constraints (this step is achieved by constraint programming).
- A new Shortest Path Algorithm to the ESPPRC since the workers's quality of service includes taking the workers' time window into consideration, and the violations are

penalised into the objective function. Classical ESPPRC does not take quality of service into account.

- A new set of instances, based on the instances of WSRP enriched with coordination constraints.

The paper is organised as follows: Section 1 is an introduction to the routing and scheduling problems with a literature review of the related works which motivate this study. Section 2 introduces the Generalised Workforce Routing and Scheduling Problem (GWSRP). Section 3 proposes the constraint-programming based decomposition method (CPDM). Section 4 presents the numerical experiments and Section 5 are concluding remarks

## 1.1 *The context of scheduling and routing problems with coordination services*

Proper coordination of services at nodes is required in different routing and scheduling problems including, but not limited to, the Workforce Scheduling and Routing Problem (WSRP), the Home Health Care (HHC), the Home Care Crew Scheduling (HCCS) and the VRP with Multiple Synchronizations (VRPMS). These problems differ mainly by the definition (or not) of connected activities and by the characteristics of workers (identical workers, or not).

Figure 1 establishes a classification of the main scheduling and routing problems, between workers' allocation problems and WPWSR. The problems are usually splited into two categories depending on coordination constraints in the routing problem.
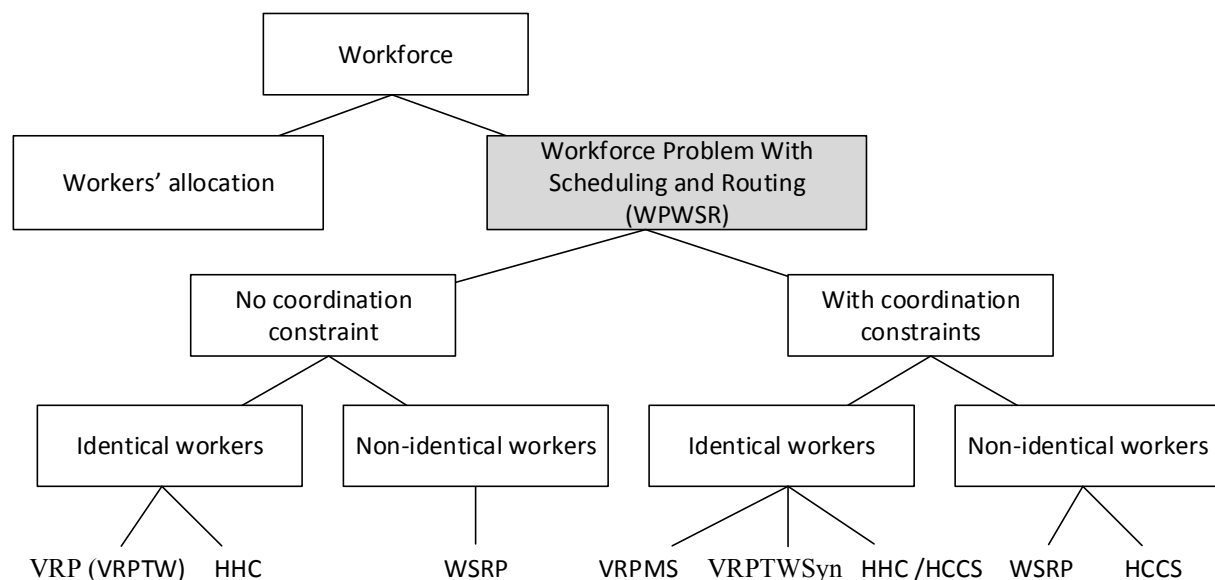


Figure 1. Classification of the main scheduling and routing problems.

Nevertheless, the boundary between these two categories is not always clearly defined and, for example, some studies addressing HHC problems do not consider coordination constraints (Bertels and Fahle, 2006), (Maenhout and Vanhoucke, 2009), (Chahed et al., 2009) while other articles take coordination constraints into account (Blais et al., 2003), (Eveborn et al., 2006), (Bachouch et al., 2011). Similar remarks hold for the WSRP without coordination constraints in (Castillo et al., 2009), (Goel and Meisel, 2013), (Laesanklang et al., 2015), (Algethami et al., 2017), (Laesanklang and Landa-Silva, 2017) and (Algethami et al., 2018) and with coordination constraints in (Laesanklang et al., 2016). Let us note that Vehicle Routing Problems (VRP) or VRP with Time-Windows (VRPTW) are commonly studied without coordination constraints, but recent studies include theses considerations, (Drexl, 2012) presents a survey of VRP with theses constraints and their applications.

Table 1 introduces the characteristics of the main routing and scheduling problems with coordination constraints considering in the most recent publications.

Table 1. Main features of routing and scheduling problems with coordination constraints.

| Problem | WSRP | VRSP-TW + TC (VRPTWSyn) | HCCS | VRPMS |
|---|---|---|---|---|
| Limited number of vehicles | X | X | X | – |
| Time window on customer | X | X | X | X |
| Time window on worker | X | X | X | – |
| Skills | X | X | X | – |
| Compatibility worker/area | X | – | – | – |
| Processing cost depending on workers | X | – | X | – |
| Compatibility customer/worker | X | X | – | – |
| **Customers' Quality of Service** | **X** | **X** | **X** | **–** |
| **Workers' Quality of Service** | **X** | **–** | **–** | **–** |
| **Non-identical workers** | **X** | **–** | **X** | **–** |
| Priority of visits | – | – | X | – |
| Vehicle capacity | – | – | – | X |
| **Objective Function** | | | | |
| Objectif 1 | Distance + processing time | Distance / Balancing / Customer QoS | Distance | Distance |
| Objectif 2 | Customers QoS | | Customers QoS | |
| Objectif 3 | **Workers QoS** | | Weighted sum of non-serviced visits | |
| Objectif 4 | Number of serviced visits | | | |

(Bredström and Rönnqvist, 2008) introduce VRSP-TW+TC (Vehicle Routing and Scheduling Problem with Time Window and Temporal Constraints) (also referred to as VRPTWSyn depending on author) which includes only pairwise synchronisation between visits. The same problem is addressed by (Liu et al., 2018) and is referred to as VRPTWSyn.

The Home Care Crew Scheduling Problem (HCCS) is a generalisation of an uncapacitated and multiple-depot VRPTW, where home carers must be assigned to visits such that the overall service level is maximised (Rasmussen et al., 2012). The service level is defined with the distance first, the processing cost (defining the customers QoS) second and the weighted sum of non-serviced visits. The distance minimisation and the processing cost minimisation are addressed successively (a strict hierarchy is defined) contrary to the WSRP where these two criteria are added to define a compromise. Five types of temporal dependencies are taken into account in HCCS including pairwise synchronisation, overlap, time lag min, time lag max, time lags min and max simultaneously (Rasmussen et al., 2012).

The VRPTW with Multiple Synchronization (VRPMS) (Hojabri et al., 2018) is an extension of the VRPTW where customers are modelled by either a regular node, or a regular node and a special node. The visits to a customer with a regular node and a special node require synchronisation. The special visit starting time is defined relatively to the regular starting time of the regular visit.

(Fink et al., 2019) introduce an Abstract Vehicle Routing Problem with Worker and Vehicle Synchronization (AVRPWVS) applied to an airport. The coordination constraints concern the travelling of workers which is done by limited capacity vehicles and the loading of baggage. Workers cannot wait at the job location (the worker must return to the depot), and jobs can be shortened by assigning additional workers. This complex problem with coordination is,

unfortunately, too far from the scheduling and routing problems considered in this paper to be taken into consideration in Table 1.

The Workforce Scheduling and Routing Problem (WSRP) involves a joint resolution of worker assignment to visits, workers schedule and trips schedule. Workers have favourite geographical areas where they want (or do not want) to work. Both customers and workers have time windows defining the customers' availability period for visits and defining the workers' availability. Customer time windows must hold whereas workers time windows and workers' favourite geographical areas can be violated. The objective function is an aggregated weighted sum of four parts: minimisation of the number of unperformed visits first; maximisation of the Quality of Service (QoS) of workers (that includes the respect of the worker's working time window and geographical working areas); maximisation of the preference satisfaction between customers and workers; and minimisation of the operational costs (which includes the transport and the cost of a worker to perform a visit). The WSRP provides an efficient modelling of the Home Health Care planning (HHC), Scheduling Technicians (ST), Security Personnel Routing and Rostering (SPRR) and Manpower Allocation (MA) as stated by (Castillo-Salazar et al., 2016). The Generalised WSRP (GWSRP) is an extension of the WSRP.

## 1.2    *Related works on WSRP*

(Castillo-Salazar et al., 2016) provide a literature survey of WSRP in an attempt to identify the common features of WSRP scenarios and the solution methods

(Castillo-Salazar et al., 2014) introduce the WSRP with a problem description, a linear formulation and a set of instances. A Steady State Genetic Algorithm (SSGA) dedicated to the WSRP is introduced by (Algethami and Landa-Silva, 2015) and specific crossover and mutation operators are defined with a direct representation of the solutions. An indirect representation is introduced by (Algethami et al., 2016) to maintain the feasibility of a solution in a Genetic Algorithm (GA). A comparative study between a SSGA (Algethami and Landa-Silva, 2015) and the Genetic Algorithm of (Algethami et al., 2016) is proposed in (Algethami et al., 2017). The Variable Neighbourhood Search (VNS) of (Pinheiro et al., 2016) takes advantage of two constructive heuristics, the first one to define a priority based list of workers per visits, and the second one to increase the workers' QoS. A clustering based approach referred to as Geographical Decomposition with Conflict Avoidance (GDCA) that splits the problem into sub-problems which consider geographical considerations and workers not performing a trip is proposed by (Laesanklang et al., 2015). The approach is extended by (Laesanklang et al., 2016) to Repeated Decomposition with Conflict Repair (RDCR) where all workers are available during the resolution of each sub-problem and may create conflicts (a worker is assigned to several trips) which are iteratively addressed. GDCA and RDCR are sub-problem resolution order dependent. The Geographical Decomposition with Conflict Repair (GDCR) (Laesanklang and Landa-Silva, 2017) also extends GDCA, the final result is not dependent on the order of sub-problem resolution.

## 2    The GWSRP: extension of the WSRP

## 2.1    *Coordination of visits*

In a large number of publications, temporal coordination constraints (also referred to as temporal dependencies) are limited to the synchronisation of visits. Since temporal coordination constraints come from real-world applications, including (but not limited to) home health care scheduling problems and scheduling technicians, let's define 9 following categories are defined. Let $i$ and $j$ define two visits, $st_i$ and $st_j$ the starting times, and $Pt_i$ and $Pt_j$ the processing times (or service times).

- C1. **Disjunctive Constraint**: several visits cannot be processed simultaneously, either $i$ is serviced first and $j$ second or the reverse: $st_i + Pt_i < st_j$ or $st_j + Pt_j < st_i$. The Disjunctive Constraint in the GWSRP takes more than two visits into consideration, and is defined to a set of visits $\vartheta$.

- C2. **Synchronisation Constraint**: $i$ and $j$ must have the same starting time ($st_i = st_j$) (Bredström and Rönnqvist, 2007; Hojabri et al., 2018), nevertheless, visits $i$ and $j$ can have different processing times.

- C3. **Minimal Duration in Common**: two visits that must have at least $T$ units of time in common (Rasmussen et al., 2012), $max(st_i, st_j) - min(st_i + Pt_i, st_j + Pt_j) \geq T$.

- C4. **Specific Time Window Constraint** $[CE; CL]$: starting time of $i$ is upper bounded by $CE$ ($st_i \leq CE$) and finishing time is lower bounded by $CL$ ($st_i + Pt_i \geq CL$).

- C5. **Conjunctive Constraint**: $j$ must start after the end of $i$, $st_i > st_j + Pt_j$.

- C6. **Conjunctive Minimal Time-Lag Delay** $TL_{ij}^{min}$ defines simultaneously: a minimal delay between the starting time of $i$ and the starting time of $j$; and that $i$ must be scheduled before $j$ (Rasmussen et al., 2012). $st_i + TL_{ij}^{min} \leq st_j$.

- C7. **Disjunctive Minimal Time-Lag Delay** defines a minimal delay between the starting times of $i$ and $j$, the value of this delay depends on the order of $i$ and $j$. $st_i + TL_{ij}^{min} \leq st_j \vee st_j + TL_{ji}^{min} \leq st_i$.

- C8. **Conjunctive Maximal Time-Lag Delay** $TL_{ij}^{max}$ defines simultaneously: a maximal delay between the starting time of $i$ and the starting time of $j$; and that $i$ must be scheduled before $j$ (Rasmussen et al., 2012) $st_i - TL_{ij}^{max} \geq st_j$ and $st_i \leq st_j$.

- C9. **Disjunctive Maximal Time-Lag Delay** defines a maximal delay between the starting times of $i$ and $j$, the value of this delay depends on the order of $i$ and $j$. $st_i \leq st_j + TL_{ij}^{max} \vee st_j \leq st_i + TL_{ji}^{max}$.

### 2.2    *A graph modelling for GWSRP*

Let us consider a graph $G = (V, E)$, where $V$ is the set of nodes, and $E$ the set of edges. $V = A \cup D \cup D'$ is composed of the set $D$ of departure locations of the workers (workers can start their trip from their own home or from a depot), the set $D'$ of finishing locations of the workers, and the set $A$ of visits characterised by their processing time and their location. The GWSRP is a WSRP problem which encompasses simultaneously constraints of categories 1 to 9.

A visit $j \in A$ models a service requiring one worker and has a processing time (or service time) $Pt_j$. Transportation time between two visits $i$ and $j$ is referred to as $T_{i,j}$ assuming that transportation times are worker independent and considering that each visit $j$ is located into a specific area.

$W$ is a set of available workers that must be assigned to visits and the process of one visit $j$ by a specific worker $w$, defines the customer's Quality of Service (QoS) referred to as $\rho_j^w \in [0; 3]$. The initial depot node of $w$ is referred to as $I_w$ and the final depot node of $w$ is referred to as $L_w$. Due to contractual commitments, the assignment of a worker $w$ to a visit $j$ may induce an extra processing cost $PrC_j^w \in \mathbb{R}$ that must be paid. $Com_j^w$ is the compatibility between a worker $w$ and a visit $j$: $Com_j^w = 0$ if worker $w$ cannot be assigned to visit $j$ and $Com_j^w = 1$ if assignment is possible ($Com_j^w$ models regulatory restrictions or compatibility skills).

The starting time of a visit $j$ is lower bounded by $E_j$ and upper bounded by $L_j$, defining the time window $[E_j; L_j]$ of visit $j$. The working time window of $w$ is $[TW_{inf}^w; TW_{sup}^w]$ and defines the

earliest and latest starting time of a visit $j$ that should be assigned to $w$, meaning that both the transportation time from the node $I_w$ to a visit $j$ and worker transportation time from $j$ to $L_w$, are not taken into account. By consequence $[TW_{inf}^w; TW_{sup}^w]$ can be referred to as the time window where visits can be processed for the worker $w$. Let us note that solutions where the worker time window constraints do not hold must be investigated since the objective function addresses the weighted number of violations of workers' availability. (Algethami and Landa-Silva, 2015).

## 2.3 Model description

The worker time window constraint is defined considering the starting time of the visit only, which means the worker can leave his depot before the lower bound of his own time window, and can return to his depot after his time window. Visits with their starting times lower or visits with their finishing times greater than the worker's time window are penalised in the objective function (since the worker's Quality of Service (QoS) is depreciated). By consequence, the starting times of the visits must not be left-shifted, but must be a compromise between the number of performed visits by a worker and the number of time his time window do not hold (to maximise the worker's quality of service).

A solution of the GWSRP is defined by an assignment of visits to workers. A sequencing of visits (defining trips) are characterised, for each visit $j$ by: arrival time $a_j$ of the worker, starting time $st_j$ of $j$, finishing time $ft_j$ of $j$, and departure time $d_j$ of the worker. Since waiting times are not taken into consideration, the finishing time of a visit is equal to the departure time of the worker. The starting time of a visit $i$ must comply with his time window $[E_j; L_j]$, the assignment of a worker to a visit has to meet data $Com_j^w = 1$ and the coordination constraints must hold.

A time window violation ($\theta_j^w$) occurs in visit $j$ performed by worker $w$, if the starting time $st_j$ of $j$ is lower than $w$'s time window ($[TW_{inf}^w; TW_{sup}^w]$): $st_j < TW_{inf}^w$; or the finishing time $ft_j$ ($ft_j = st_j = Pt_j$) of $j$ is greater than $w$'s time window $ft_j > TW_{sup}^w$. All the constraints, except the coordination constraints, are compliant with the WSRP which is fully described in (Algethami and Landa-Silva, 2017) and (Garaix et al., 2018).

Binary variables introduced by (Algethami and Landa-Silva, 2017) are the following:

$x_{i,j}^w$ $\begin{cases} =1 \text{ if worker } w \text{ moves from } i \text{ to } j \\ =0 \text{ otherwise} \end{cases}$

$\theta_j^w$ $\begin{cases} =1 \text{ if time window violation occurs} \\ \quad \text{when } j \text{ is assigned to worker } w \\ =0 \text{ otherwise} \end{cases}$

$\psi_j^w$ $\begin{cases} =1 \text{ if worker } w \text{ is assigned to a visit } j \\ \quad \text{with } PA_j^w = 0 \\ =0 \text{ otherwise} \end{cases}$

$y_j$ $\begin{cases} =1 \text{ if } j \text{ is not performed} \\ =0 \text{ otherwise} \end{cases}$

The objective function $f$ to be minimised incorporates the key aspects that make a high-quality solution: low operational costs and high Quality of Service (QoS). The objective function includes four criteria balanced by four weights ($\lambda_1, \dots, \lambda_4$) and the associated weight to each criteria corresponds to a priority level and clearly defines a hierarchy between the criteria (see (Algethami and Landa-Silva, 2017)) with $\lambda_1 \ll \lambda_2 \ll \lambda_3 \ll \lambda_4$. It is important to note that $\theta_j^w$ depends on the starting time of the visits. The objective function may be outlined by:

$$f = \lambda_1(Routing + Scheduling\ costs)$$
$$+ \lambda_2(Customers'\ QoS) + \lambda_3(Workers'\ QoS) + \lambda_4(Unserviced\ visits)$$

The formal definition of the objective function is:

$$f = \lambda_1 \sum_{w=1}^{W} \sum_{i=0}^{A} \sum_{j=1}^{A} \left( T_{i,j} + PC_j^w \right) x_{i,j}^w + \lambda_2 \sum_{j=1}^{A} \sum_{i=0}^{A} \sum_{w=1}^{W} (3 - \rho_j^w) \, x_{i,j}^w + \lambda_3 \sum_{j=1}^{A} \sum_{w=1}^{W} \left( \psi_j^w + \theta_j^w \right)$$

$$+ \lambda_4 \sum_{j=1}^{A} y_j$$

## 3    Proposition: a constraint-programming based decomposition method

The constraint-programming based decomposition method (CPDM) defines a method which takes advantage of: a column generation based scheme which has been proven to be efficient in routing problems for decades and a method working on trips to allow intensive local search tailored to the routing to produce slight trip modifications by a constraint programming solver. The CPDM consists of 3 steps (Figure 2):

- **Initial trip construction** (Step 1, section 3.1) is based on a column generation scheme with a time limit of `SCGlimitTime`, where coordination constraints are not addressed. The slave problem is a specific Elementary Shortest Path Problem with Resource Constraints (ESPPRC), taking workers' time window violations into consideration. This step returns a set $TGC$ of trips.

- **Selection of trips** solves a set partitioning problem, which maximises the number of coordination constraints taken into account and that minimises the cost (Step 2, section 3.2). This step returns: a set $t$ of trips, with $|t| = |W|$ where $t$ contains one trip per worker (the trip can be empty); a set $ChCC$ of checked coordination constraints; and a set $UnCC$ of coordination constraints that do not hold.

- **Reconstruction of trips** with coordination constraints based on a Constraint-Programming (CP) model. This step (Step 3, section 3.3) generates a solution with coordination constraints by slight modifications of the initial trips. This step iterates until a maximal number of iterations `iter_max` is reached and is called the Iterative Constraint Programming based Algorithm (ICPA). ICPA is an iterative approach that tries to find a solution ensuring all coordination constraints belonging to $C_l = ChCC \cup c$, and where the set $C_l$ contains all coordination constraints belonging to $ChCC$ and a constraint $c \in UnCC$.
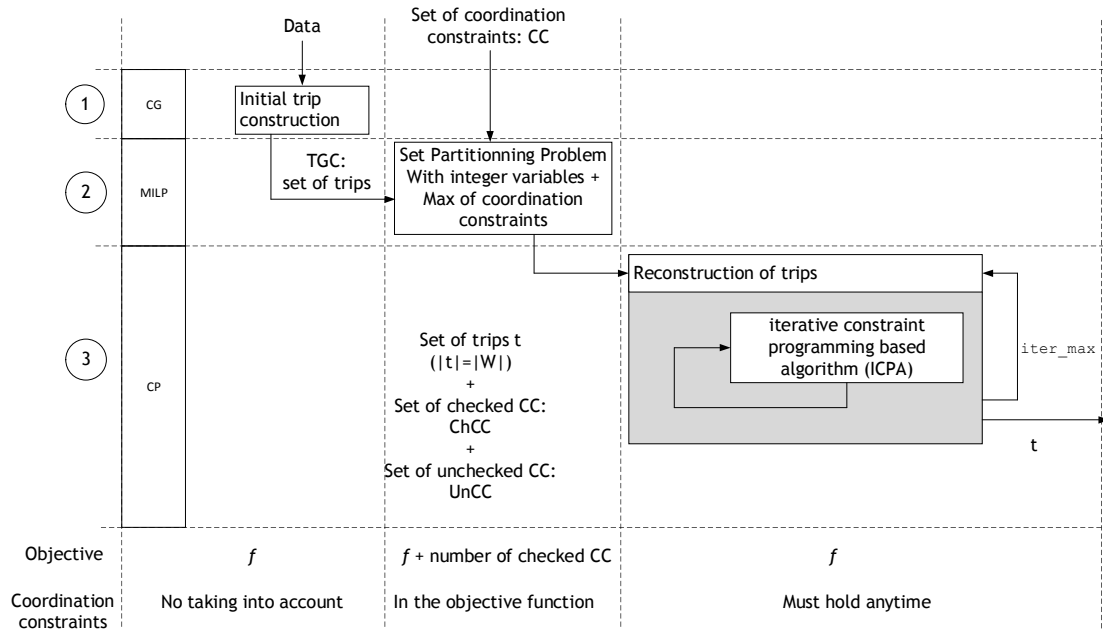
Figure 2. Constraint-Programming based Decomposition Method.

Algorithm 1 is the pseudo-code for the CPDM approach. The first step is `Initial_Trip_Construction()` which is the column generation (line 13). Line 14 is the selection of trips (`Selection_of_trip()`) and line 16 is the iterative reconstruction of trips with a CP approach. Each step is explained in the following sections.

**Algorithm 1**. constraint-programming based decomposition method

```
1.  procedure CPDM
2.  input parameters
3.    SCGlimitTime: time limit for the column generation
4.    SClimitTime:  time limit for the set partitioning problem
5.    iter_max:     maximal number of iterations for the
6.                  reconstruction phase
7.    Data:         data of the instance
8.    C:            set of coordination constraints
9.  output parameters
10.   t: set of trips
11.   Cost: solution cost
12. begin
13.    TCG = Call Initial_Trip_Construction(Data)          // Step 1
14.    (t,C*,C') = Call Selection_of_trips(TGC)            // Step 2
15.    for i=1 to iter_max do                              // Step 3
16.      (cost,c*,C',t)= Call Reconstruction_of_trips(t,C*,C')
17.    endfor
18. end
```

### 3.1 A column generation scheme for the initial trip construction (Step 1)

The first step of the CPDM is the computation of initial trips without any consideration of the coordination constraints. This step is based on a column generation (CG) scheme since GC is well adapted to a WSRP without coordination constraints. The slave problem is solved by a specific Elementary Shortest Path Problem with Resource Constraints (ESPPRC) (section 3.1.2) algorithm.

A column $k$ represents a trip to a worker $w$ and it is characterised by:

$CT_k$        cost of trip $k$

$B_k^j$ $\qquad$ = 1 if visit $j$ is achieved during trip $k$
$\qquad\qquad$ = 0 otherwise

$E_k$ $\qquad$ = $w$, is the worker performing trip $k$

The cost $CT_k$ of a trip $k$ is positive and depends on the worker who has been assigned, on the set of visits serviced by the worker, and on the starting times of the visits.

$$CT_k = \lambda_1 \sum_{i \in A \cup I_w} \sum_{j \in A \cup L_w} ( T_{i,j} + PC_j^w)x_{i,j}^w + \lambda_2 \sum_{j \in A} \sum_{i \in A \cup I_w} (3 - \rho_j^w) x_{i,j}^w + \lambda_3 \sum_{j \in A} ( \psi_j^w + \theta_j^w)$$

### 3.1.1 Master problem: a set partitioning problem

The master problem combines the workers' trips (or columns) to minimise the total cost of the trips and to maximise the number of performed visits. A solution of the Master Problem is a selection of columns (or trips), minimising the sum of the cost of the trips, such that: (C1) each visit $j$ is performed exactly once and (C2) each worker achieves at most one trip. A binary variable $y_k$ determines if a column $k$ is chosen ($y_k = 1$) or not ($y_k = 0$). The set $\Omega$ of columns is initialised with the identity basis. A trip on the identity basis starts at the depot, visits only one visit $j$ and returns back to the depot. This trip is assigned to a fictitious worker $w_0$ and the trip cost is defined considering $\lambda_4$ i.e. the cost of an unassigned visit.

The master problem is equivalent to a set partitioning problem.

$$f_{master} = Min \ \textstyle\sum_{k \in \Omega} CT_k \ y_k$$

$$s.t. \qquad \forall j \in A \qquad\qquad \sum_{k \in \Omega} B_k^j \ y_k = 1 \qquad\qquad (C1)$$

$$\forall w \in W \qquad\qquad \sum_{k \in \Omega | E_k = w} y_k \le 1 \qquad\qquad (C2)$$

Let $\alpha_j$ denote the dual value related to the master problem constraint (C1) of visit $j$, and $\beta_w$ the dual value related to the master problem constraint (C2) concerning worker $w$.

### 3.1.2 Slave problem: a new Elementary Shortest Path Problem with Resource Constraints

The slave problem is related to a worker $w$ and consists of computing a valid trip for a worker $w$, minimising the reduced cost (also referred to as the objective function $f_{slave}$) which depends on: 1) the cost of the trip $CT_k$; 2) the dual prices $\alpha_j$ of the visits; and 3) the dual price $\beta_w$ of $w$. It is important to keep in mind that the cost of the trip $CT_k$ depends on the starting times of the visits.

$$f_{slave} = \lambda_1 \sum_{i \in A \cup I_w} \sum_{j \in A \cup L_w} ( T_{i,j} + PC_j^w)x_{i,j}^w + \lambda_2 \sum_{j \in A} \sum_{i \in A \cup I_w} (3 - \rho_j^w) x_{i,j}^w + \lambda_3 \sum_{j \in A} ( \psi_j^w + \theta_j^w)$$
$$- \sum_{j \in A} \sum_{i \in A \cup I_w} \alpha_j x_{ij}^w - \beta_w$$

**Classical Dynamic Programming (DP) to solve the ESPPRC**

The slave problem can be efficiently solved using a dedicated Elementary Shortest Path Problem with Resource Constraints (ESPPRC) procedure extending the classical label algorithms which have been the focus of numerous publications (Feillet et al., 2004),(Irnich and Desaulniers, 2005),(Irnich and Villeneuve, 2006), (Righini and Salani, 2006), (Lozano and Medaglia, 2013), (Lozano et al., 2016), (Thomas et al., 2019).

Briefly, a partial path starting at the depot node $I_w$ is iteratively extended to nodes until the trip reaches the final depot node $L_w$. A partial path (also referred to as, for convenience, a solution or a partial solution), on node $i$, is modelled by a label $l_i$ which is extended along the arcs going out of $i$, to obtain new labels. The extension of each label is restricted to nodes ensuring an elementary feasible trip. A label $l_i = (CT, f_{slave}, B, d_j, st_j, to\_visit)$ represents a partial or final solution where: $CT$ is the trip cost; $f_{slave}$ is the reduced cost; $B$ is a binary vector of performed visits ($B_j = 1$ if visit $j$ is achieved during the trip, otherwise $B_j = 0$); $d_j$ is the earliest arrival time of the worker on visit $j$; $st_j$ is the starting time of the worker on visit $j$; and $to\_visit$ is the list of visits which can be reached by the worker from the current label.

**A new Dynamic Programming to solve the ESPPRC with workers' time window violations**

As the violation of a worker's time window is allowed (variables $\theta_j^w$ in the objective functions), several situations must be investigated to define the starting time of visit $j$. As a reminders, a worker's time window violation occurs when either the worker starts/ finishes the visit before/after his own time window. If worker $w$ arrives on visit $j$ during, or after his working time window ($da_j \in [TW_v^-; TW_v^+]$), then a unique label must be considered with the earliest starting time of $j$. If the worker arrives at visit $j$ before his working time window ($da_j \leq TW_v^-$), two cases have to be investigated: either the worker starts as early as possible the visit and a penalty is added to the reduced cost ($CT$) of the trip (case 1 on Figure 3); either the worker starts during his working time window and no penalty is applied (case 2 on Figure 3), but if the worker starts later than during the previous case, a lower number of visits might be performed during his trip. These two cases are separately processed, and for each case, a new label is created. These considerations lead to a new extended version of the ESPPRC algorithm.
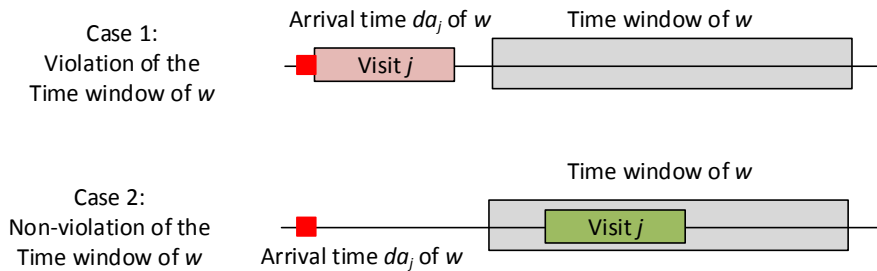


Figure 3. Two possibilities to the starting time of the visit.

### 3.2 Selection of trips: set partitioning formulation with maximisation of coordination constraints (Step 2)

The Column Generation (CG) (step 1 of the CPDM) is solved to the root node, and to obtain an integer solution, avoiding a costly branch-and-price heuristic, a set partitioning formulation is solved at the root node on the set $TGC$ of columns found by the CG. The set partitioning problem includes integer constraints and a hierarchical function where 1) the number of coordination constraints that can be taken into account is maximised first 2) the solution cost $f$ is minimised second.

The objective function of the set partitioning formulation is:

$$f^{spp} = \Lambda_1 \max(f^v) + \Lambda_2 \min(f)$$

where $\Lambda_1$ and $\Lambda_2$ guarante that $Min(\Lambda_1 \max(f^v)) > Max(\Lambda_2 \min(f))$. For example, $\Lambda_2 = 1$ and $\Lambda_1 = 1000$ can be compliant values that permit to obtain readable $f^{spp}$. $f^v = \sum_{c \in CC} VC^c$,

$CC$ is the set of all coordination constraints, $VC^c$ is a binary variable, which values 0 if constraint $c \in CC$ is checked and values 1 otherwise.

The set partitioning formulation is given by:

$$\forall j \in A \qquad \sum_{k \in TGC} B_k^j \, y_k = 1 \qquad\qquad (C1')$$

$$\forall w \in W \qquad \sum_{k \in TGC \mid E_k = w} y_k \leq 1 \qquad\qquad (C2')$$

$$\forall t \in TGC, \forall j \in t \qquad st_j + Pt_j + T_{j,j+1} \leq st_{j+1} \qquad\qquad (C3)$$

$$\forall c \in CC \qquad VC^c = \begin{cases} = 0 \text{ if } c \text{ holds} \\ = 1 \text{ otherwise} \end{cases} \qquad\qquad (C4)$$

All trips found during the step 1 are feasible, and they are not modified during this second step: only starting times of the visits are computed, and are potentially updated to ensure that coordination constraints hold (constraint $C3$).

At the end of this procedure, a subset $UnCC$ of coordination constraints that have not been addressed during resolution ($c \in UnCC \mid VC^c = 1$) and a subset $ChCC$ of coordination constraints which hold3 remain.

### 3.3 Reconstruction of trips (Step 3)

Since not all coordination constraints are addressed due to step 1, and as the solution found is heuristic due to step 2, the objective of the last step of the CPDM is to improve the trips to ensure all coordination constraints hold firstly and to find a better cost solution secondly. This third step is based on a CP model since the capabilities of the CP solver are exploited to further restrict the search space and quickly find a feasible solution. The reconstruction of trips is similar to a local search operator focusing on trips with the objective of providing improvements in a short computational time.

#### 3.3.1 Reconstruction phase with coordination constraints method

The objective of the reconstruction phase with coordination constraints is to slightly modify the current trips obtained by the step 2 of the CPDM, in order to find a solution which takes the coordination constraints into consideration. The reconstruction phase is iterated several times and it is based on the Iterative Constraint Programming based Algorithm (ICPA) that iteratively investigate new trips with a CP solveur.

#### 3.3.2 An iterative constraint programming based algorithm

The aim of the Iterative Constraint Programming based Algorithm (ICPA) is to search for a solution addressing all coordination constraints and minimising the cost, with an iterative approach which is related to a local search (Algorithm 3).

The overall procedure is introduced in Algorithm 2. The starting point of the ICPA is an initial solution (the trips $t$), a set $ChCC$ of checked constraints and a set $UnCC$ of unchecked constraints. The ICPA tries to find a solution by inserting iteratively constraints belonging to $UnCC$. $C^{ICPA}$ is the subset of current constraints taken into consideration at each iteration, $C^{ICPA}$ is initialised with $C^{ICPA} = SCC\_Ch$ (line 13). An unchecked coordination constraint $c \in UnCC$ is added to $C^{ICPA}$ at each iteration (line 17). If a new solution is found (line 19), this solution becomes the new initial solution (line 21). If the CP resolution fail, the last constraint added to the CP model is not added into set $ChCC$. In both cases, the process iterates with the next coordination constraint.

Experimentally, the ICPA approach has been found more efficient when the CP solver is stopped after the given time limit `SPClimitTime`. If ICPA has not found a solution after this time limit, it is unlikely that it will find one if it runs longer. This approach leads to a shorter running time and therefore the selection of the variables in the branching tree is a key point as variables selected at the beginning of the search will not be reconsidered in the limited time. During the first call of ICPA, the first variables selected in the branching tree are all variables $a_i$, concerning the assignment of visit $i$ to a worker; and then all the variables $s_i$, concerning the successor of $i$ (these variables are presented in detail section 3.3.3). During the next call of ICPA, first variables selected in the branching tree are $s_i$, and then $a_i$. At each new call, the order of $a_i$ and $s_i$ in the branching tree is inverted. Moreover, among the set of variables $a_i$ (or $s_i$), the order of selection of these variables is randomly generated to explore a large solution space, some numerical experiments and discussions are presented in section 4.4.2.

**Algorithm 2**. iterative constraint programming based algorithm (ICPA)

```
1.   procedure ICPA()
2.   input parameters
3.     t: set of trips
4.     UnCC: set of constraints not addressed
5.     ChCC: set of addressed constraints
6.   output parameters
7.     t: set of trips
8.     Cost: solution cost
9.     UnCC: set of constraints not addressed
10.    ChCC: set of addressed constraints
11.  begin
12.    S = random_order_of_var_in_CP()
13.    C^ICPA = ChCC
14.    (t',Cost, new_sol)= solve_CP(t, C, S)
15.    for ∀c ∈ UnCC do
16.      S = random_order_of_var_in_CP()
17.      C^ICPA = ChCC ∪ {c}
18.      (t',Cost, ok)= solve_CP(t, C, S)
19.      if(new_sol)then
20.        ChCC = C^ICPA
21.        UnCC = UnCC − {c}
22.        t = t'
23.      endif
24.    endfor
25.  end
```

### 3.3.3 The Constraint-Programming model

A visit $j$ for a vehicle routing problem is usually defined by four variables (Cordeau and Laporte, 2003): the starting time $st_j$, the finishing time $df_j$, the arrival time $da_j$ of the worker, and the departure time $dd_j$ of the worker as shown on Figure 4 where $j$ is preceded by visit $i$. The departure time $dd_j$ is equal to the finishing time $df_i$ since no waiting time consideration is taken into account. A worker in the GWSRP is equivalent to a vehicle in a VRP.
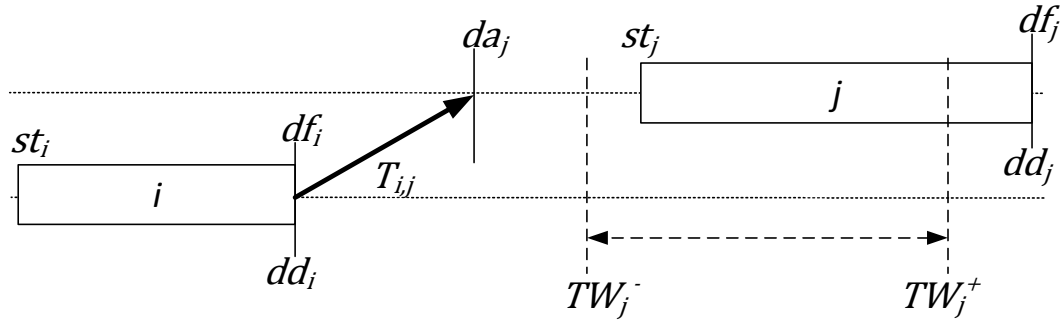
Figure 4. Definition of a trip.

The CP-based formulation of the problem is based on the following set of data considering the propositions of (Hojabri et al., 2018; Rossi et al., 2006) in terms of notation and model.

Data:

$t$: set of initial trips.

$C^{ICPA}$: set of coordination constraints to ensure.

$V = A \cup D \cup D'$ : set of visits ($A$), departure locations ($D$) and final locations ($D'$).

Variables:

$a_i$: the assignment of visit $i$ to a worker.

$s_i$: the successor of visit $i$.

$p_i$: the position of visit $i$ in the worker's trip performing $i$.

$pred_i$: predecessor of visit $i$.

$st_i$: the starting time of visit $i$.

$df_i$: the finishing time of visit $i$.

$da_i$: the arrival time of the worker to visit $i$.

$dd_i$: the departure time of the worker from visit $i$.

$$\theta_i = \begin{cases} =1 \text{ if worker time window violation} \\ \quad \text{occurs when the worker visits } i \\ =0 \text{ otherwise} \end{cases}$$

$$\psi_i = \begin{cases} =1 \text{ if } i \text{ is assigned to a worker } w \\ \quad \text{such that } PA_j^w = 0 \\ =0 \text{ otherwise} \end{cases}$$

$$y_i = \begin{cases} =1 \text{ if visit } i \text{ is not performed} \\ =0 \text{ otherwise} \end{cases}$$

Domains of the variables:

$\forall i \in V$      $s_i \in V$
$\forall i \in V$      $a_i \in W$
$\forall i \in V$      $p_i \in [0; |A|]$
$\forall i \in V$      $pred_i \in V$
$\forall i \in A$      $da_i \in [0; D_{max}]$
$\forall i \in A$      $st_i \in [TW_i^-; TW_i^+]$
$\forall i \in A$      $df_i \in [TW_i^- + Pt_i; TW_i^+ + Pt_i]$
$\forall i \in A$      $dd_i \in [0; D_{max}]$
$\forall i \in A$      $\theta_i \in \{0; 1\}$
$\forall i \in A$      $\psi_i \in \{0; 1\}$

Constraints:

The CP model extends the model of (Hojabri et al., 2018) for constraint (1) to (5), and encompasses constraints (6) − (9) to take into account the temporal constraints of trips including the starting time $st_i$ of visit $i$ and the worker arrival time $da_i$. Two visits cannot have the same successor, which is prohibited by constraint (1) (AllDifferent). Constraint (2) states a visit and its successor must be performed by the same worker. Constraints (3) and (4) avoid subtours

(Pesant et al., 1998). Constraint (5) is a link between the predecessor and successor to reinforce the model.

$$\text{AllDifferent}(s_{i|i\in N}) \tag{1}$$

$$\forall i \in V \qquad\qquad a_i = a_{s_i} \tag{2}$$

$$\forall i \in V \qquad\qquad s_i \neq i \tag{3}$$

$$\forall i \in A \cup D \qquad\qquad p_{s_i} = p_i + 1 \tag{4}$$

$$\forall i \in V \qquad\qquad (pred_k = i) \Leftrightarrow (s_i = k) \tag{5}$$

The constraints (6) – (9) define the temporal constraints considering the visit time windows. As the waiting time is not weighted, a solution with $dd_i = df_i$, and $da_{s_i} = dd_i + T_{ij}$ is required, as illustrated in Figure 4.

$$\forall i \in A \cup D \qquad\qquad da_{s_i} = dd_i + T_{i,s_i} \tag{6}$$

$$\forall i \in V \qquad\qquad st_i \geq dda_i \tag{7}$$

$$\forall i \in V \qquad\qquad df_i = st_i + Pt_i \tag{8}$$

$$\forall i \in V \qquad\qquad dd_i = df_i \tag{9}$$

Constraints (10) – (12) define the WSRP constraints: (10) ensures only an authorised worker can perform the visit. If a visit starts before the time window $[TW_{inf}^w; TW_{sup}^w]$ of the assigned worker $w$, a penalty is considered with $\theta_i = 1$, similarly a penalty is taken into account if visit $i$ finishes out of worker $w$'s time window (11). This constraint is difficult to express without using a "or" constraint since the left member if not continuous. A judicious implementation in order to obtain an efficient model is to use a global constraint $DiffN$ (Beldiceanu and Contejean, 1994). Constraint (12) ensures that a penalty is taken into consideration according to the worker preference area (depending on visit $i$).

$$\forall i \in A | Com_j^w = 0, \forall w \in W \qquad a_i \neq w \tag{10}$$

$$\forall i \in A \qquad\qquad st_i \notin [TW_{inf}^{a_i}; TW_{sup}^{a_i} - Pt_i] \Leftrightarrow \theta_i = 1 \tag{11}$$

$$\forall i \in A \qquad\qquad PA_i^{a_i} = 0 \Rightarrow \psi_i = 1 \tag{12}$$

The objective function $f$ is the classical weighted GWSRP (or WSRP) objective function.

$$f = \lambda_1 \sum_{i \in N \cup V^d} T_{i,s_i} + \lambda_1 \sum_{i \in N} PC_i^{a_i} + \lambda_2 \sum_{i \in N} \rho_i^{a_i} + \lambda_3 \sum_{i \in N} (\theta_i + \psi_i) + \lambda_4 \sum_{i \in N} (y_i)$$

The coordination requirements are addressed by constraints (14) – (22). Constraint (14) prohibits execution of disjunctive visits, belonging to the set $\vartheta$ in $E_{c1}$, with the global constraint DiffN() (category 1). Constraint (15) imposes the same starting time to synchronised visits (category 2). Constraint (16) requires a minimal duration $T$ in common between $i$ and $j$ (category 3). A specific time window (category 4) is defined by constraint (17). Constraint (18) ensures the relative order between $i$ and $j$ (category 5). Constraint (19) defines a conjunctive minimal time-lag constraint i.e, the starting time of $j$ ($st_j$) is upper bounded by $st_i + TL_{ij}^{min}$ and the relative order between $i$ and $j$ is defined by $st_i \leq st_j$. Constraint (20) models the disjunctive minimal time-lag delay (category 7) between two visits. Constraint (21) models conjunctive maximal time-lag delay (category 8) i.e. they define a visit $j$ time window relative to the starting time of $i$. Constraint (22) concerns the disjunctive maximal time-lag delay (category 9).

$$\forall i \in \vartheta \ (\vartheta \in E_{c1}) \qquad\qquad \text{DiffN}(st_i) \tag{14}$$

$$\forall (i,j) \in E_{c2} \qquad\qquad st_i = st_j \tag{15}$$

$$\forall (i,j) \in E_{c3} \qquad\qquad st_j \leq st_i + p_i - T \wedge st_j \geq st_i + T - p_j \tag{16}$$

$$\forall i \in E_{c4} \qquad\qquad CL \leq st_i + pt_i \wedge st_i \leq CE \tag{17}$$

$$\forall i \in E_{c5} \qquad\qquad st_i + Pt_i \leq st_j \tag{18}$$

$$\forall (i,j) \in E_{c6} \qquad st_i + TL_{ij}^{min} \leq st_j \qquad\qquad (19)$$
$$\forall (i,j) \in E_{c7} \qquad st_i + TL_{ij}^{min} \leq st_j \vee st_j + TL_{ji}^{min} \leq st_i \qquad (20)$$
$$\forall (i,j) \in E_{c8} \qquad st_j \leq st_i + TL_{ij}^{max} \wedge st_i \leq st_j \qquad (21)$$
$$\forall (i,j) \in E_{c9} \qquad st_i \leq st_j + TL_{ij}^{max} \vee st_j \leq st_i + TL_{ji}^{max} \qquad (22)$$

As the CP looks for one local reconstruction of trips in an iterative search process, additional refinements are introduced to improve the model. Parameters $\delta$ and $\gamma$ control the maximal distance with, respectively, the assignment and the successors which are defined by the initial solution. If too large values are used for both $\delta$ and $\gamma$, CP solver spends time evaluating solutions strongly different of the current ones whereas too small values will lead CP solver to fail in trip modifications to address the new coordination constraints. Let us define $A_i$ and $S_i$ the assignment and the successor of visit $i$ in the set of trips $t$ at the start of ICPA. Let us define two binary variables $ba_i$ and $bs_i$ such that $ba_i = a_i \neq A_i$ and $bs_i = s_i \neq S_i$. Both distances are defined by $\sum_{\forall i \in V} ba_i \leq \delta$ and $\sum_{\forall i \in V} bs_i \leq \gamma$.

## 4    Computational evaluation for the GWSRP

To the best of our knowledge, no instance is available and dedicated to the GWSRP considering the 9 coordination constraints introduced in section 2.1.

Firstly, the method is tested on the 30 benchmark instances introduced by (Bredström and Rönnqvist, 2007) who define a VRPTWSyn that encompasses only synchronisation constraints. Secondly, a new set of instances is introduced and dedicated to the GWSRP. These instances are based on (Castillo-Salazar et al., 2016)'s instances to the WSRP.

### 4.1    Parameters of the experiments

Table 2 introduces speed factors that are used in the following comparative tables to ensure a fair comparative study and note that the speed factor has been established according to research articles including (Dongarra, 2014), as well information taken from http://www.roylongbottom.org.uk/ linpackresults.htm and from http://asteroidsathome.net/boinc/cpu_list.php.

Since MIPS performance is not the only parameter that impact the CPU time, Table 2 also provides the information available about the computer, the operating system and the language. It is important to note that for some articles, not enough information is given to compute the speed factor.

Table 2. Relative performances of computers.

| | (Bredström and Rönnqvist, 2007) | | (Bredström and Rönnqvist, 2008) | (Afifi et al., 2016) | (Liu et al., 2018) | |
|---|---|---|---|---|---|---|
| **Name** | **BP1** | **BP2** | **H** | **SA-ILS** | **ALNS** | **CPDM** |
| **Nb of Runs** | **1** | | **1** | 10 | **Xx** | 1 |
| **Computer** | Intel Xeon 2.67GHz | | Intel Xeon 2.67GHz | Intel Xeon 2.67 GHz | Intel E5-2670 | Intel Core i7-4790 3.60 GHz |
| **OS** | | | | LINUX | LINUX | Windows |
| **Language** | C++ | | C++ | C++ | C++ | C++/Java |
| **MFlops** | / | | / | / | 2570 | 2671 |
| **Ratio** | / | | / | / | 0.96 | 1 |

The parameters values in Table 3 have been adjusted empirically by considering a subset of large instances and are identical in all instances except for the instances GGLT-13 to GGLT-17 due to the large scale of these instances. The linear models are solved by CPLEX 12.7 using

a single thread (Parallel mode is disabled) and the constraint programming solver used is Choco 4.0.9. The column generation is coded in C++.

Table 3. CPDM parameters.

| | | | (Bredström and Rönnqvist, 2007)'s instances | GGLT 1 to 12 | GGLT 13-17 |
|---|---|---|---|---|---|
| **CPDM** | Initial trip construction (Column Generation) | SCGlimitTime | 3600 seconds | 3600 seconds | |
| | | Labels/node | 250 | 250 | 50 |
| | Selection of trip | NbThreadsCPLEX | 1 | 1 | |
| | | SClimitTime | 150 sec | 150 sec | 1260 sec |
| | reconstruction phase with coordination constraints | iter_max | 15 | 15 | |
| | | TimeLimit/Iter | 10 sec | 10 sec | 40 sec |
| | | Branchement | 50% A / 50% S | 50% A / 50% S | |
| | | $\delta$ | N/2 | N | |
| | | $\gamma$ | N/4 | N | |
| **CPLEX resolution** | | NbThreadsCPLEX | / | 1 | |
| | | TimeLimit | / | 10800 sec | |

## 4.2 *Benchmarks for the instances of (Bredström and Rönnqvist, 2007)*

The VRPTWSyn introduced by (Bredström and Rönnqvist, 2007) can be solved by the GWSRP model, nevertheless the VRPTWSyn only takes synchronisation constraints into consideration. This comparative study concerns five dedicated methods to VRPTWSyn and the CPDM which is not dedicated to the VRPTWSyn. The objective is to evaluate the capability of CPDM to provide quality solutions and to prove the method has some merits as regards dedicated methods.

Table 4 introduces the results to each instance of the five methods: two branch-and price (BP1 and BP2) proposed by (Bredström and Rönnqvist, 2007), a heuristic H from (Bredström and Rönnqvist, 2008), a heuristic SA-ILS proposed by (Afifi et al., 2016) and an ALNS introduced by (Liu et al., 2018). The second column of Table 4 is the best known solutions, and, in bold the solutions which proved to be optimal. $T$ is the computational time to find the best solution.

The experiments in Table 4, prove that the CPDM has its merits in providing quality solutions for all instances from VRPTWSyn but cannot compete with the dedicated method of (Liu et al., 2018) that provides results very close to the optimal solutions in a very short computational time. The (Bredström and Rönnqvist, 2007)'s instances and the CPDM solution can be downloaded at: http:\\www.isima.fr\~lacomme\GWSRP\.

Table 4 highlights the number of instances solved by the 6 methods, the average time and the average gap. Although, the CPDM is not dedicated to the VRPTWSyn, the method is able to find a solution to each instance which is not the case for BP1, BP2 and H methods. The SA-ILS and the ALNS have a very low computational time, but it is the best time on several runs and so no fair comparative study can hold with the total computational time of the CPDM.

Table 4. CPDM average results on VRPTWSyn instances.

| | Number of instances solved | Avg. T (seconds) | Avg Gap (%) |
|---|---|---|---|
| (Bredström and Rönnqvist, 2007) – BP1 | 27 | 1598 | 1.4 |
| (Bredström and Rönnqvist, 2007) – BP2 | 29 | 1479 | 0.8 |
| (Bredström and Rönnqvist, 2008) – H | 17 | 374 | 3.2 |
| (Afifi et al., 2016) – SA-ILS | 30 | 31 | 0.2 |
| (Liu et al., 2018) – ALNS | 30 | 4 | 0.0 |
| Our proposition – CPDM | 30 | 136 | 3.9 |

### 4.3  Benchmarks for the GWSRP

Considering that the GWSRP is being addressed for the first time, a set of 17 instances has been generated with $|A| = 32 \ldots 177$ visits and $|W| = 2 .. 59$ workers (Table 5). The total number of coordination constraints (TTC, last column on Table 5) varies from 3 for instance number 4 to 49 for instance number 15. To constraint $C1$, one disjunctive constraint may encompass several visits, hence the number of binary disjunctive constraints is given by the column "C1 Nb disjunctions".

These instances come from the (Castillo-Salazar et al., 2016) original instances which have been turned into GWSRP instances. Test instances were created from WSRP instances in such a way that solution trips obtained by the relaxation of coordination constraints and minimising the cost, provided poor quality lower bounds and were strongly different from the trips used in a quality (near optimal) solutions. The test instances can be found at the following web page: http:\\www.isima.fr\~lacomme\GWSRP\.

Table 5. Main features of the GWSRP Instances.

| Instance | Number of visits | Number of workers | C1 | C1 Nb disjunctions | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | TTC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 32 | 5 | 3 | 19 | 3 | 1 | 2 | 3 | 3 | 1 | 2 | 1 | 19 |
| 2 | 31 | 6 | 4 | 8 | 1 | 3 | 0 | 3 | 1 | 2 | 0 | 1 | 15 |
| 3 | 38 | 7 | 6 | 28 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 9 |
| 4 | 28 | 8 | 3 | 45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 5 | 13 | 2 | 1 | 6 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 6 |
| 6 | 28 | 4 | 2 | 21 | 1 | 0 | 0 | 0 | 2 | 2 | 0 | 3 | 10 |
| 7 | 36 | 9 | 3 | 32 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 |
| 8 | 71 | 34 | 7 | 123 | 3 | 5 | 0 | 3 | 7 | 8 | 1 | 5 | 39 |
| 9 | 30 | 10 | 3 | 67 | 2 | 1 | 0 | 2 | 1 | 2 | 1 | 2 | 14 |
| 10 | 62 | 16 | 7 | 163 | 2 | 0 | 0 | 1 | 1 | 2 | 0 | 2 | 15 |
| 11 | 57 | 17 | 4 | 244 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| 12 | 61 | 21 | 2 | 20 | 2 | 1 | 0 | 1 | 2 | 2 | 1 | 2 | 13 |
| 13 | 177 | 55 | 6 | 106 | 3 | 2 | 0 | 1 | 2 | 2 | 3 | 4 | 23 |
| 14 | 50 | 19 | 5 | 70 | 5 | 3 | 0 | 2 | 4 | 5 | 4 | 7 | 35 |
| 15 | 198 | 54 | 25 | 320 | 3 | 0 | 0 | 2 | 3 | 10 | 1 | 5 | 49 |
| 16 | 150 | 54 | 15 | 250 | 6 | 1 | 0 | 3 | 2 | 9 | 1 | 3 | 40 |
| 17 | 170 | 59 | 12 | 279 | 2 | 2 | 0 | 2 | 5 | 7 | 1 | 3 | 34 |

Table 6 introduces the results found on 17 GWSRP instances which were solved by a linear formulation with CPLEX and by the proposed approach CPDM. $S$ is the value of the best solution found, $Gap\ CPLEX$ is the gap of CPLEX between the best integer solution found and the fractional solution, $T^*$ is the time to find the best solution, $TT$ is the total time, $Gap(S)$ is the gap (in percentage) between the best solution found by CPLEX and the best solution found by CPDM approach. The last line is the average value of the computational time and the average value of $Gap(S)$.

The time limit to the CPLEX resolution is 10 800 seconds (see Table 3), and the optimal solution is not found for instances 8, 12, 13, 14, 15, 16 and 17. The CPDM is able to find the optimal solution to instances 5, 6 and 7 ($Gap(S) = 0\%$); and a strongly high quality solution to instances 1, 2, 3, 4, and 9 ($Gap(S) <= 1\%$). For the instances 8 and 10, the gap is less than 6%. Only the instance 11 has an important gap ($Gap(S) = 14.7\%$). For the instances 12, 13, 14, 15, 16 and 17 solutions that is better than the CPLEX solutions is found ($Gap(S) < 0\%$). Concerning instances 15, 16 and 17, CPLEX is not able to find a solution where all visits are serviced in 10 800 seconds, whereas the CPDM is able to find one in less than 40 000 seconds. Hence, the gap is very important ($Gap(S) < -99\%$). On average, the $Gap(S)$ is about $-19.1\%$.

Except for instance 1, the total computational time of the CPDM is strictly lower than the CPLEX resolution. For example, for instance number 4, the CPLEX computational time is 6 372.8 seconds, and is only 106.4 seconds for the CPDM, which is 60 times lower. The average computational time of the CPDM is approximately 6 times lower than the CPLEX resolution (902.3 seconds on average for the CPDM and 5 678.5 seconds on average for the CPLEX resolution). The typical question that arises is whether the solution gap between the CPDM and the CPLEX resolution could be reduced by increasing the number of iterations, or computational time. Several experiments have been carried out but no significant improvement have been obtained.

Table 6. Solutions found on the GWSRP instances.

| Instance | CPLEX | | | CPDM | | | |
|---|---|---|---|---|---|---|---|
| | S | Gap CPLEX % | TT (sec) | S | T* (sec) | TT (sec) | Gap(S)% |
| 1 | **703.3** | 0.0 | 174.9 | 705.0 | 220.5 | 238.5 | 0.2 |
| 2 | **207.2** | 0.0 | 489.4 | 208.6 | 31.6 | 109.6 | 0.7 |
| 3 | **886.5** | 0.0 | 1636.8 | 889.7 | 162.4 | 162.4 | 0.4 |
| 4 | **516.8** | 0.0 | 6372.8 | 518.5 | 28.4 | 106.4 | 0.3 |
| 5 | **106.6** | 0.0 | 55.2 | **106.6** | 1.5 | 1.5 | **0.0** |
| 6 | **1033.5** | 0.0 | 800.1 | **1033.5** | 166.5 | 166.5 | **0.0** |
| 7 | **558.1** | 0.0 | 8061.4 | **558.3** | 155.9 | 203.9 | **0.0** |
| 8 | 645.2 | 0.3 | 10811.3 | 660.4 | 704.1 | 746.1 | 2.4 |
| 9 | **135.7** | 0.0 | 842.5 | 136.2 | 49.1 | 109.1 | 0.4 |
| 10 | **490.4** | 0.0 | 1110.3 | 517.0 | 813.5 | 867.5 | 5.4 |
| 11 | **410.9** | 0.0 | 978.8 | 471.2 | 455.5 | 521.5 | 14.7 |
| 12 | 414.5 | 2.6 | 10804.1 | **412.2** | 210.3 | 276.3 | **−0.6** |
| 13 | 7576.3 | 37.5 | 11113.0 | **6875.1** | 1958.4 | 1982.4 | **−9.3** |
| 14 | 1552.0 | 69.2 | 10801.6 | **943.9** | 302.0 | 338.0 | **−39.2** |
| 15 | 700412.0 | 99.9 | 10836.3 | **1057.8** | 3892.7 | 3916.7 | **−99.8** |
| 16 | 421222.0 | 99.9 | 10834.3 | **1254.3** | 2413.8 | 2461.8 | **−99.7** |
| 17 | 590418.0 | 99.9 | 10811.4 | **1107.1** | 2938.9 | 3131.0 | **−99.8** |
| AVG | | | 5 678.5 | | 853.2 | 902.3 | −19.1 |

Table 7 presents the details of each solution found, the four criteria of the objectif function are mde explicit. Coefficients associated with each criteria are: $\lambda_1 = 0.1$, $\lambda_2 = 10.0$, $\lambda_3 = 100.0$ and $\lambda_4 = 10\,000.0$. Table 7 highlights that the CPDM is able to find the same value at criteria C4 and C3 as the CPLEX solution. These two criteria are the most weighted. For instance 13, the CPDM find a better criteria C3 than CPLEX (21 vs 29). Let us note the instances 1, 4, 5, 6, and 7 where the criteria C2 is identical to the CPLEX and CPDM solution, but where C1 is higher to the CPDM solution (522 vs 462 for instance 1, for example). Instance 2 is interesting since CPDM has a lower criteria C1 (508) than the CPLEX (524), but the criteria C2 of CPDM is the highest (15.78 vs 14.48). A reversal of the situation occurs to instance 9 where the CPDM solution has a lower criteria C2 than the CPLEX solution (10.19 vs 10.29), but the highest criteria C1 (343 vs 328). Concerning instance 12, the CPDM is able to find a solution with both criteria C1 and C2 lower than the CPLEX solution.

For the instances 15, 16 and 17, the CPLEX resolution is not able to find a solution where all visits are serviced ($C4 > 0$); whereas CPDM resolution finds solutions where all visits are serviced ($C4 = 0$).

Table 7. Detail of GWSRP solutions.

| Instance | CPLEX | | | | | CPDM | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | C1 ($\lambda_1$) | C2 ($\lambda_2$) | C3 ($\lambda_3$) | C4 ($\lambda_4$) | Total | C1 ($\lambda_1$) | C2 ($\lambda_2$) | C3 ($\lambda_3$) | C4 ($\lambda_4$) | Total |
| 1 | 462 | 55.71 | 1 | 0 | **703.3** | 479 | 55.71 | 1 | 0 | 705.0 |
| 2 | 524 | 15.48 | 0 | 0 | **207.2** | 508 | 15.78 | 0 | 0 | 208.6 |
| 3 | 715 | 41.5 | 4 | 0 | **886.5** | 737 | 41.6 | 4 | 0 | 889.7 |
| 4 | 263 | 29.05 | 2 | 0 | **516.8** | 280 | 29.05 | 2 | 0 | 518.5 |
| 5 | 163 | 9.03 | 0 | 0 | **106.6** | 163 | 9.03 | 0 | 0 | 106.6 |
| 6 | 577 | 27.58 | 7 | 0 | **1033.5** | 577 | 27.58 | 7 | 0 | 1033.5 |
| 7 | 860 | 27.21 | 2 | 0 | **558.1** | 862 | 27.21 | 2 | 0 | 558.3 |
| 8 | 1278 | 31.74 | 2 | 0 | 645.2 | 1284 | 33.2 | 2 | 0 | **660.4** |
| 9 | 328 | 10.29 | 0 | 0 | **135.7** | 343 | 10.19 | 0 | 0 | 136.2 |
| 10 | 1097 | 28.07 | 1 | 0 | **490.4** | 1088 | 30.82 | 1 | 0 | 517.0 |
| 11 | 876 | 32.33 | 0 | 0 | **410.9** | 906 | 38.06 | 0 | 0 | 471.2 |
| 12 | 961 | 31.93 | 0 | 0 | 414.5 | 935 | 31.87 | 0 | 0 | **412.2** |
| 13 | 4227 | 425.36 | 29 | 0 | 7576.3 | 4855 | 428.96 | 21 | 0 | **6875.1** |
| 14 | 2166 | 53.54 | 8 | 0 | 1552.0 | 2221 | 62.18 | 1 | 0 | **943.9** |
| 15 | 1371 | 27.5 | 0 | 70 | 700412 | 5771 | 48.07 | 0 | 0 | **1057.8** |
| 16 | 3060 | 31.59 | 6 | 42 | 421222 | 5513 | 70.3 | 0 | 0 | **1254.3** |
| 17 | 1502 | 26.78 | 0 | 59 | 590418 | 4970 | 61.01 | 0 | 0 | **1107.1** |

Table 8 introduces the solution found by CPLEX where the resolution is limited by the CPDM running time for each instance. For example, for instance number 2, the resolution time of the CPDM is 109.6 seconds, therefore, CPLX resolution time is limited to 110 seconds. The gap is the gap between the CPDM solution and the CPLEX solution. For instance number 2, the gap is positive ($Gap(S) > 2.6\%$) meaning that CPDM solution has a better cost than the CPLEX solution: 208.6 for the CPDM solution and 214.1 for the CPLEX solution. For instances number 10 and 11, the $Gap(S)$ is negative meaning the CPLEX solution is better than the CPDM solution for the same running time. For instances 12, 13, 14, 15, 16 and 17 the $Gap(S)$ is close to 99% since the CPLEX is not able to find a solution where all visits are serviced. For example, for instance 15, the best solution found by CPLEX leads to 70 visits that are not serviced for a cost of 412 (the objective function is a hierarchic function that can be written for readibility 70/0412).

This study highlights the abilities of the CPDM to find a better solution on average (36.4%) than the CPLEX resolution during the same time limit.

Table 8. Comparison between CPLEX and CPDM with the same computational time

| Instance | CPLEX | | | CPDM | | | |
|---|---|---|---|---|---|---|---|
| | S | Gap CPLEX % | TT (sec) | S | T* (sec) | TT (sec) | Gap(S)% |
| 1 | 703.3 | 0.0 | 153.1 | 705.0 | 220.5 | 238.5 | -0.2 |
| 2 | 214.1 | 11.0 | 110.1 | 208.6 | 31.6 | 109.6 | **2.6** |
| 3 | 898.7 | 2.5 | 162.1 | 889.7 | 162.4 | 162.4 | **1.0** |
| 4 | 517.3 | 0.6 | 106.2 | 518.5 | 28.4 | 106.4 | -0.2 |
| 5 | 108.8 | 4.8 | 2.0 | 106.6 | 1.5 | 1.5 | **2.0** |
| 6 | 1043.5 | 10.6 | 167.1 | 1033.5 | 166.5 | 166.5 | **1.0** |
| 7 | 558.4 | 30.9 | 204.1 | 558.3 | 155.9 | 203.9 | **0.0** |
| 8 | 1203.6 | 47.1 | 747.2 | 660.4 | 704.1 | 746.1 | **45.1** |
| 9 | 135.8 | 5.6 | 109.1 | 136.2 | 49.1 | 109.1 | -0.3 |
| 10 | 490.4 | 0.1 | 867.6 | 517.0 | 813.5 | 867.5 | -5.4 |
| 11 | 433.2 | 6.1 | 522.6 | 471.2 | 455.5 | 521.5 | -8.8 |
| 12 | 21008.5 | 98.1 | 277.6 | 412.2 | 210.3 | 276.3 | **98.0** |
| 13 | 48596.4 | 90.3 | 1986.5 | 6875.1 | 1958.4 | 1982.4 | **85.9** |
| 14 | 500000.0 | 99.9 | 338.4 | **943.9** | 302.0 | 338.0 | **99,8** |
| 15 | 650780.0 | 99.9 | 3930.0 | **1057.8** | 3892.7 | 3916.7 | **99,8** |
| 16 | 451196.0 | 99.9 | 2468.4 | **1254.3** | 2413.8 | 2461.8 | **99,7** |
| 17 | 123363.0 | 99.6 | 3140.0 | **1107.1** | 2938.9 | 3131.0 | **99,1** |
| AVG | | | | | | | **36.4** |

### *4.4 Sensitivity analysis regarding the insertion order of synchronised constraints and the branching order*

The CPDM and, in particular, the ICPA depend on the insertion order of synchronised constraints and on the branching order. Other orders could lead to another and possibly better solution. The objective of the following sections is to study the impact of both the insertion order and the branching order. These experiments are conducted with instance 10L where at the end of Step 2 (selection of trips), 8 constraints are not yet checked.

#### 4.4.1 Insertion order

During Step 3, an order of insertion is randomly selected and in order to evaluate the impact of the insertion sequence of synchronised constraints, a comparative study is achieved focusing on this sequence. The branching order of the variables in the CP tree is always the same, and 10 random sequences of insertions of synchronised constraints are generated.

Table 9 introduces the numerical results of this experiment. The number of unchecked constraints is 8 at the beginning of the ICPA, and this number is 0 at the end of each run, meaning that each time the ICPA is able to find a solution encompassing all coordination constraints. The standard deviation of the cost is 8.4. In conclusion, the order of the insertion of the synchronisation constraints does not have an important effect on the results.

Table 9. Consequence of the insertion order on synchronised constraints.

| Run | Nb of unchecked constraints at the start of ICPA | Nb of unchecked constraints at the end of ICPA | S |
|---|---|---|---|
| 1 | 8 | 0 | 670 |
| 2 | 8 | 0 | 654 |
| 3 | 8 | 0 | 670 |
| 4 | 8 | 0 | 654 |
| 5 | 8 | 0 | 654 |
| 6 | 8 | 0 | 654 |
| 7 | 8 | 0 | 670 |
| 8 | 8 | 0 | 654 |
| 9 | 8 | 0 | 670 |
| 10 | 8 | 0 | 670 |
| **Standard Deviation** | | **0** | **8.4** |

#### 4.4.2 Branching order

This second experiment concerns the impact of the branching order of variables in the CP search tree, on the instance 10L, during the ICPA. The sequence of insertions of synchronised constraints is the same for the 10 runs, but the branching order of variables is randomly generated.

Table 10 provides the number of unchecked constraints at the end of the ICPA and the cost of the solutions. Runs 2, 4, 5 and 6 do not provide a solution where all synchronised constraints are holding. The standard deviation is 0.5. The solution cost varies in the interval $[610; 694]$, 610 for run 4 and 694 for run 5, the standard deviation is 30.4.

Table 10. Consequence of the branching order in the CP search tree.

| Run | Nb of unchecked constraints at the start of ICPA | Nb of unchecked constraints at the end of ICPA | S |
|---|---|---|---|
| 1 | 8 | 0 | 655 |
| 2 | 8 | 1 | 687 |
| 3 | 8 | 0 | 624 |
| 4 | 8 | 1 | 610 |
| 5 | 8 | 1 | 694 |
| 6 | 8 | 1 | 632 |
| 7 | 8 | 0 | 672 |
| 8 | 8 | 0 | 611 |
| 9 | 8 | 0 | 648 |
| 10 | 8 | 0 | 669 |
| **Standard Deviation** | | **0.5** | **30.4** |

This experiment highlights the importance of the branching order in the CP search tree on the quality of the solution, and points out that an iterative approach with different orders of branching is advantageously compared to a unique run.

## 5    Concluding remarks

A WSRP variant, referred to as GWSRP, incorporating new complex coordination constraints between visits is introduced in this paper. The objective function of the GWSRP includes operational costs, quality of service of customers, quality of service of workers and number of serviced visits. The GWSRP is modelled as a graph. Due to the quality of service to workers, a solution of the GWSRP is not semi-active. A new evaluative function is introduced to find the starting time of visits maximising the quality of service to workers without coordination constraint.

The GWSRP is solved by a Constraint-Programming based Decomposition Method (CPDM) which includes a three-step process. The first one is a column generation without coordination constraints, based on the new evaluative function, solved to the root node. The second step is the resolution of a set partitioning problem in order to obtain an integer solution. The third step is the reconstruct of trips by adding iteratively the coordination constraints, using a CP model.

Numerical experiments on instances derived from (Castillo-Salazar et al., 2016)'s original instances demonstrated the capacity of the CP to be used as a powerful local search method in a decomposition approach. This proposition is in the same trend of research as (Hojabri et al., 2018), and highlight the efficiency of CP included in an iterative approach.

The numerical evaluations establish that the proposed approach is able to solve integrated scheduling and routing problem with coordination constraints and to obtain high quality solutions.

## References:

Afifi, S, Duc-Cuong, D, Moukrim, A. 2016. "Heuristic solutions for the vehicle routing problem with time windows and synchronized visits". Optimization Letters 10: 511–525.

Algethami, H, Landa-Silva, D. 2017. "Diversity-based adaptive genetic algorithm for a Workforce Scheduling and Routing Problem", in: Evolutionary Computation (CEC), 2017 IEEE Congress On. IEEE. 1771–1778.

Algethami, H, Landa-Silva, D, 2015. "A Study of Genetic Operators for the Workforce Scheduling and Routing Problem" MIC 2015: The XI Metaheuristics International Conference.

Algethami, H, Landa-Silva, D, Martínez-Gavara, A. 2017. "Selecting Genetic Operators to Maximise Preference Satisfaction in a Workforce Scheduling and Routing Problem", in: Proceedings of the 6th International Conference on Operations Research and Enterprise Systems. Presented at the ICORES, Porto, Portugal, 416–423.

Algethami, H, Martínez-Gavara, A, Landa-Silva, D, 2018. "Adaptive multiple crossover genetic algorithm to solve workforce scheduling and routing problem". Journal of Heuristics. 1-40.

Algethami, H., Pinheiro, R.L., Landa-Silva, D., 2016. "A genetic algorithm for a workforce scheduling and routing problem". IEEE Congress on Evolutionary Computation (CEC). 927-934.

Attia, E.-A, Duquenne, P, Le-Lann, J-M, 2014. "Considering skills evolutions in multi-skilled workforce allocation with flexible working hours". International Journal of Production Research 52: 4548–4573.

Bachouch, R.B, Guinet, A, Hajri-Gabouj, S, 2011. "A Decision-Making Tool for Home Health Care Nurses". Planning. Supply Chain Forum: An International Journal 12:14–20.

Beldiceanu, N, Contejean, E, 1994. "Introducing global constraints in CHIP". Mathematical and Computer Modelling 20: 97–123.

Bertels, S, Fahle, T, 2006. "A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem". Computers & Operations Research 33: 2866–2890.

Blais, M, Lapierre, S.D, Laporte, G, 2003. "Solving a home-care districting problem in an urban setting". Journal of the Operational Research Society 54: 1141–1147.

Bredström, D, Rönnqvist, M, 2008. "Combined vehicle routing and scheduling with temporal precedence and synchronization constraints". European Journal of Operational Research 191: 19–31.

Bredström, D, Rönnqvist, M, 2007. "A Branch and Price Algorithm for the Combined Vehicle Routing and Scheduling Problem With Synchronization Constraints". SSRN Electronic Journal.

Castillo, I, Joro, T, Li, YY, 2009. "Workforce scheduling with multiple objectives". European Journal of Operational Research 196: 162–170.

Castillo-Salazar, J.A, Landa-Silva, D, Qu, R. 2016. "Workforce scheduling and routing problems: literature survey and computational study". Annals of Operations Research 239: 39–67.

Castillo-Salazar, J.A, Landa-Silva, D, Qu, R. 2014. "Computational Study for Workforce Scheduling and Routing Problems". ICORES. 434–444.

Chahed, S, Marcon, E, Sahin, E, Feillet, D, Dallery, Y. 2009. "Exploring new operational research opportunities within the Home Care context: the chemotherapy at home". Health Care Management Science 12: 179–191.

Chankov, S, Hütt, M-T, Bendul, J.. 2018. "Influencing factors of synchronization in manufacturing systems". International Journal of Production Research 56: 4781–4801.

Cordeau, J.-F, Laporte, G. 2003. "A tabu search heuristic for the static multi-vehicle dial-a-ride problem". Transportation Research Part B: Methodological 37: 579–594.

Dongarra, J.J. 2014. "Performance of various computers using standard linear equations software". Report CS-89-85, University of Manchester.

Drexl, M. 2012. "Synchronization in Vehicle Routing—A Survey of VRPs with Multiple Synchronization Constraints". Transportation Science 46: 297–316.

Eveborn, P, Flisberg, P, Rönnqvist, M. 2006. "Laps Care—an operational system for staff planning of home care". European Journal of Operational Research 171: 962–976.

Feillet, D, Dejax, P, Gendreau, M, Gueguen, C. 2004. "An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems". Networks 44: 216–229.

Fink, M, Desaulniers, G, Frey, M, Kiermaier, F, Kolisch, R, Soumis, F. 2019. "Column generation for vehicle routing problems with multiple synchronization constraints". European Journal of Operational Research 272: 699–711.

Garaix, T, Gondran, M, Lacomme, P, Mura, E, Tchernev, N. 2018. "Workforce Scheduling Linear Programming Formulation". IFAC-PapersOnLine 51: 264–269.

Goel, A, Meisel, F. 2013. "Workforce routing and scheduling for electricity network maintenance with downtime minimization". European Journal of Operational Research 231, 210–228.

Hojabri, H, Gendreau, M, Potvin, J-Y, Rousseau, L-M. 2018. "Large neighborhood search with constraint programming for a vehicle routing problem with synchronization constraints". Computers & Operations Research 92: 87–97.

Irnich, S, Desaulniers, G. 2005. "Shortest Path Problems with Resource Constraints", in: Desaulniers, G, Desrosiers, J, Solomon, M.M (Eds.), Column Generation. Springer-Verlag, New York, 33–65.

Irnich, S, Villeneuve, D. 2006. "The Shortest-Path Problem with Resource Constraints and k - Cycle Elimination for k = 3". INFORMS Journal on Computing 18: 391–406.

Koochaki, J, Bokhorst, J.A.C, Wortmann, H, Klingenberg, W. 2013. "The influence of condition-based maintenance on workforce planning and maintenance scheduling". International Journal of Production Research 51: 2339–2351.

Laesanklang, W, Landa-Silva, D. 2017. "Decomposition techniques with mixed integer programming and heuristics for home healthcare planning". Annals of Operations Research 256, 93–127.

Laesanklang, W, Landa-Silva, D, Castillo-Salazar, J.A. 2016. "An investigation of heuristic decomposition to tackle workforce scheduling and routing with time-dependent activities constraints" International Conference on Operations Research and Enterprise Systems: 239–260.

Laesanklang, W, Pinheiro, R.L, Algethami, H, Landa-Silva, D. 2015. "Extended decomposition for mixed integer programming to solve a workforce scheduling and routing problem". ICORES: 191–211.

Liu, R, Tao, Y, Xie, X. 2018. "An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits". Computers & Operations Research 101.

Lozano, L, Duque, D, Medaglia, A.L. 2016. "An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints". Transportation Science 50: 348–357.

Lozano, L, Medaglia, A.L. 2013. "On an exact method for the constrained shortest path problem. Computers & Operations Research 40: 378–384.

Maenhout, B, Vanhoucke, M. 2009. "The impact of incorporating nurse-specific characteristics in a cyclical scheduling approach". Journal of the Operational Research Society 60: 1683–1698.

Mathlouthi, I, Gendreau, M, Potvin, J.-Y. 2018. "Mixed integer linear programming for a multi-attribute technician routing and scheduling problem". INFOR 56: 33–49.

Mundschenk, M, Drexl, A. 2007. "Workforce planning in the printing industry". International Journal of Production Research 45: 4849–4872.

Pesant, G, Gendreau, M, Potvin, J.-Y, Rousseau, J.-M. 1998. "An Exact Constraint Logic Programming Algorithm for the Traveling Salesman Problem with Time Windows". Transportation Science 32: 12–29.

Pinheiro, R.L, Landa-Silva, D, Atkin, J. 2016. "A Variable Neighbourhood Search for the Workforce Scheduling and Routing Problem". Advances in Nature and Biologically Inspired Computing: 247–259.

Rasmussen, M.S, Justesen, T, Dohn, A, Larsen, J. 2012. "The Home Care Crew Scheduling Problem: Preference-based visit clustering and temporal dependencies". European Journal of Operational Research 219: 598–610.

Righini, G, Salani, M. 2006. "Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints". Discrete Optimization 3: 255–273.

Rossi, F, van Beek, P, Walsh, T. 2006. "Handbook of Constraint Programming". Elsevier.

Saadat, M, Tan, M.C.L, Owliya, M, Jules, G. 2013. "Challenges and trends in the allocation of the workforce in manufacturing shop floors". International Journal of Production Research 51: 1024–1036.

Suter, E, Deutschlander, S, Makwarimba, E, Wilhelm, A, Jackson, K, Lyons, S.W, 2014. "Workforce utilization in three continuing care facilities". Health Sociology Review 23: 65–76.

Thomas, B.W, Calogiuri, T, Hewitt, M. 2019. "An exact bidirectional A ? approach for solving resource-constrained shortest path problems". NETWORKS 73: 187–205.

Wirojanagud, P, Gel, E.S, Fowler, J.W, Cardy, R. 2007. "Modelling inherent worker differences for workforce planning". International Journal of Production Research 45: 525–553.

Wongmongkolrit, S, Rassameethes, B. 2010. "An algorithm of workforce scheduling for maintenance optimization". International Journal of Management Science and Engineering Management 5: 163–169.

Xu, Z, Ming, X.G, Zheng, M, Li, M, He, L, Song, W. 2015. "Cross-trained workers scheduling for field service using improved NSGA-II". International Journal of Production Research 53: 1255–1272.