An algorithm proposal for a minimum cost SDR multi-standard system using Graph Theory

Patricia KAISER * #, Yves LOUËT *, Amine EL SAHILI #

* *IETR/SUPELEC, Campus de Rennes*
Avenue de la Boulaie, CS 47601, F-35576 Cesson-Sévigné cedex, France
patricia.kaiser@supelec.fr, yves.louet@supelec.fr

Mathematic Department, Lebanese University
Hadath, Beirut, Lebanon
sahili@ul.edu.lb

***Abstract*— The design of future multi-standard systems remains a challenge due to increasing flexibility requirements. Promising solutions include designing flexible radio architectures that exploit common aspects between the different set of standards cohabiting in the device. In this paper, graph theory appears and particularly the study of directed hypergraphs, which helps in the research concerning minimum cost multi-standard designs. A cost function which calculates the cost of any possible option of implementation is mentioned but its derivations won't be in the scope of this paper. Our objective is to optimize this proposed cost function to its minimum possible value and thus solving the optimization problem that finds balance between flexibility and computing efficiency. For this, we propose a Minimum Cost Design (MCD) algorithm capable of selecting the option which has the minimum cost to pay and will present its complete set of instructions in this paper. This algorithm exploits various definitions and notations of directed hypergraphs.**

***Index Terms*— Directed Hypergraphs, Software-Defined Radio, graph structure, minimum cost design.**

I. INTRODUCTION

Numerous standards have evolved and still are rapidly evolving for various wireless applications. Consequently, radio system designers must develop flexible equipment that support different existing standards and can no longer focus their attention on one standard. Therefore, the flexible Software-Defined Radio (SDR) concept [1] is emerging as a potential solution for developing a device which dynamically adapts to the radio environment, by replacing conventional radio hardware with reconfigurable, reprogrammable radios.

The possibilities to design software radio architectures range from the "Velcro" approach to the "Very Fine Grain" approach. The former approach aims to support several communication standards through dedicated self-contained complex communication components, while the latter is based on manipulating small size operators to support different standards. However, a promising approach to realize an SDR multi-standard terminal is to identify the appropriate common functions and operators between and inside the standards. This is what's called the "parametrization" approach [2].

An approach for designing flexible multi-standard radio systems is proposed, which consists in exploring such designs at different levels of granularity and selects the convenient level depending on each designer's needs. This is translated into a graph structure of the multi-standard system in [3], which describes the interrelationships between the different components in the system. This graph representation provides all the options of implementation capable of realizing the multi-standard system. However, a cost function which calculates the cost of each of these options is introduced in [4].

In our work, we exploit the theory of graphs to model and characterize various aspects of the SDR multi-standard system. Graph theory [5] is the study of graphs used to model pairwise relations between objects from a certain collection. A "graph" in this context refers to a collection of vertices and a collection of edges that connect pairs of vertices. A graph may be undirected, meaning that there is no distinction between the two vertices associated with each edge, or its edge may be directed from one vertex to another in which case it is called a digraph. Hypergraphs [6] and directed hypergraphs [7] are generalizations of graphs and digraphs respectively.

The area of our interest is to optimize the proposed cost function to its minimum value possible. This is a problem related to determining an optimal or near-optimal resource sharing for multi-standard systems which is faced with a complex cost/objective function. All exact methods known for determining an optimal solution require a computing effort which increases exponentially with number of nodes, so that in practice exact solutions can be attempted only on problems involving fewer nodes. Thus, near-optimal techniques were selected in [3] including the Simulated Annealing (SA) algorithm [8] and Genetic Algorithm (GA) [9] which can find acceptable good solutions rather than the best possible solution (applied on examples of our optimization problem), in less computing time.

In this paper, we present a new algorithm that finds the minimum cost option capable of implementing an SDR multi-standard system. It explores the different options of implementation, excluding some particular options proved inefficient for the search of a minimum cost design in [10]. This algorithm, on the contrary to SA and GA, is capable of providing an exact optimal solution. Besides, it's a new idea algorithm which exploits various definitions and notations of directed hypergraphs.

The rest of this paper is organized as follows. After the present section, some basic required definitions of directed hypergraphs are reported in section 2. In the subsequent two sections, we briefly present the graph modeling of the SDR multi-standard system and a suggested cost function respectively, using graph theory. In section 5, we introduce the MCD algorithm whose role is to select an option of implementation which has the minimum cost and that exploits various notations of directed hypergraphs. Finally, a conclusion's section ends this paper.

II. Definitions of Directed Hypergraphs Notions

A. Hypergraphs

A hypergraph H is defined by a pair $H = (V, E)$, where $V = \{x_1, x_2, \cdots, x_n\}$ is the set of vertices of H and $E = \{e_1, e_2, \cdots, e_m\}$, with $e_i \subseteq V$, $e_i \neq \phi$ for $i = 1, 2, \cdots, m$, denotes the set of hyperedges of H. Clearly, when each edge e_i contains exactly two vertices of V, the hypergraph is a standard graph [6].

B. Directed hypergraphs

A directed hypergraph is a hypergraph but with directed hyperedges (also called hyperarcs) where a directed hyperedge e is an ordered pair $e = (X, Y)$ of (possibly empty) disjoint subsets of vertices; X is the tail of e denoted by $T(e)$ and Y is its head denoted by $H(e)$ [7]. Fig. 1 is an example of a directed hypergraph $H = (V, E)$ such that:

$V = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$
$E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ where:
$e_1 = (\{x_1, x_2\}, \{x_3\})$
$e_2 = (\{x_3\}, \{x_7\})$
$e_3 = (\{x_3, x_4\}, \{x_5, x_6\})$
$e_4 = (\{x_7\}, \{x_1\})$
$e_5 = (\{x_1, x_7\}, \{x_8\})$
$e_6 = (\{x_9\}, \{x_8\})$

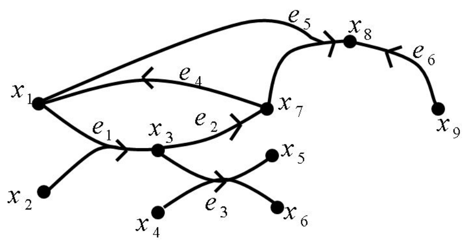

Fig. 1. A directed hypergraph $H = (V, E)$

C. Forward and Backward star

Let $H = (V, E)$ be a directed hypergraph and let $v \in V$. The Forward Star and the Backward Star of node v are defined by:
$FS(v) = \{e_i \in e, v \in T(e_i)\}$ and
$BS(v) = \{e_i \in e, v \in H(e_i)\}$ respectively [7].
As an example, the forward and backward stars of node x_3 in Fig. 1 are respectively:
$FS(x_3) = \{e_2, e_3\}$ *and* $BS(x_3) = \{e_1\}$.

D. Paths in directed hypergraphs

In a directed hypergraph $H = (V, E)$, a path from r to n $(r, n \in V)$ is defined by a sequence of nodes and hyperarcs $P = (v_1 = r, e_{i1}, v_2, e_{i2}, v_3, \cdots, e_{iq}, v_{q+1} = n)$ such that: $r \in T(e_{i1})$, $n \in H(e_{iq})$ and $v_j \in H(e_{i(j-1)}) \cap T(e_{ij})$ $j = 2, \cdots, q$

This path is called an rn-path, where r is its origin and n is its destination. The length of a path is equal to the number of hyperarcs traversed along. Indeed, we'll denote $V(P)$ and $E(P)$ by the set of vertices and hyperarcs respectively traversed via P. For example, in Fig. 1, we have $Q = (x_2, e_1, x_3, e_2, x_7, e_5, x_8)$ is a path from x_2 to x_8 of length 3, where $V(Q) = \{x_2, x_3, x_7, x_8\}$ and $E(Q) = \{e_1, e_2, e_5\}$ [7].

E. BF-reductions of hyperarcs

Let $e = (T(e), H(e))$ be a hyperarc in a directed hypergraph H. A BF-reduction of e is a hyperarc $(\{x\}, \{y\})$ where $x \in T(e)$ and $y \in H(e)$.
For example, we have $(\{x_3\}, \{x_5\})$, $(\{x_3\}, \{x_6\})$, $(\{x_4\}, \{x_5\})$, $(\{x_4\}, \{x_6\})$ are all BF-reductions of the hyperarc e_3 in Fig. 1.

F. Weight of a path

Let $P = P_{rn} = (v_1 = r, e_{i1}, v_2, e_{i2}, v_3, \cdots, e_{iq}, v_{q+1} = n)$ be an rn-path.
Let $e_{ij} \in E(P)$. We'll define the BF-reduction of e_{ij} via the path P by its particular BF-reduction obtained by selecting the predecessor vertex to e_{ij} in the path P as its specific tail node and the successor vertex to e_{ij} in P as its head node. Denote $BF_P(e_{ij})$ by this BF-reduction of e_{ij} via P. Then according to the definition, we get: $BF_P(e_{ij}) = (\{v_j\}, \{v_{j+1}\})$ $j = 1, 2, \cdots, q$.
Suppose that we have a directed hypergraph $H = (V, E)$ in which a positive integer weight is assigned to every BF-reduction of any hyperarc in E. For every P a path between r and n, we'll denote the weight of P by the product of the weights of the BF-reductions via P of all the hyperarcs in $E(P)$. So we can write:

$$w(P) = \prod_{e_{ij} \in E(P)} w(BF_P(e_{ij})), \tag{1}$$

where $w(P)$ denotes the weight of the path P and $w(BF_P(e_{ij}))$ stands for the weight of $BF_P(e_{ij})$ in H.

G. Sub-Hypergraphs

Let H be a directed hypergraph. We say that X is a *sub-hypergraph* of H if it satisfies that:
$V(X) \subseteq V(H)$ & $E(X) \subseteq E(H)$, where $T(e), H(e) \subseteq V(X)$ $\forall e \in E(X)$.
Note that the forward and backward stars of node v in X will be defined as:

- $FS_X(v) = \{e \in E(X)/v \in T(e)\}$
- $BS_X(v) = \{e \in E(X)/v \in H(e)\}$

Let X be a sub-hypergraph of H s.t $E(X) \neq E(H)$ and let $e \in E(H)$ but e doesn't belong to $E(X)$.
Define the sub-hypergraph X' of H obtained from X and e such that: $V(X') = V(X) \cup H(e) \cup T(e)$ and $E(X') = E(X) \cup \{e\}$. Then X' is called a sub-hypergraph of H induced by X and e, and will be denoted by $X + e$.

H. G-path with root M

Let H be a directed hypergraph and $M \subseteq V(H)$.
A G-path X of H with root M is defined by a sub-hypergraph of H satisfying:

1) $|FS_X(u)| = 0$ or 1 $\forall\, u \in V(X)$
2) $M \subseteq V(X)$
3) $\forall\, u \in V(X)$, $\exists$ a path from v to u for some $v \in M$

III. Representation for SDR system implementation using graph theory

In this section, we first explore a model for multi-standard systems as a graph with different levels of granularity, which enables to select the convenient levels depending on each designer's needs. Afterwards, we provide a theoretical representation of the graph structure of the multi-standard system as a directed hypergraph. Finally, we briefly present a suggestion concerning a graphical representation of any selected option of implementation.

A. Graph modeling of SDR systems

The multi-standard reconfigurable system was represented as a graph with many different layers (or levels) in [3]. Each Processing Element (PE) occupies a certain layer depending upon its granularity level, where more complex PEs have higher granularity levels than less complex ones that can form their functionalities. The goal of this approach is to provide the options to the designer, to select a set of operators, each of them at the most appropriate

level of granularity, as dictated by his needs.
A graph structure of two standards S & T is presented in Fig. 2. Each node represents an elementary PE. In order to perform the functionalities of this PE, it can be installed by itself in the design, as a unified nondivisible block, or it can be realized by some lower-level building blocks. A node of a higher level, called a parent node, may have dependencies with nodes of underlying levels, called descendant nodes. Two node dependencies occur between nodes of different levels. An "OR" dependency (direct arrow), as the hyperarc between $A2$ & $B1$ in Fig. 2, means that one descendant node ($B1$) called several specific times is necessary to implement the parent node ($A2$). However, an "AND" dependency ("inverted Y" connection), as the hyperarc between $A2$ and $B2$ & $B3$ in Fig. 2, signifies that all descendant nodes via the "AND" dependency ($B2$ & $B3$) are needed to implement the parent node ($A2$) accompanied with certain number of calls.

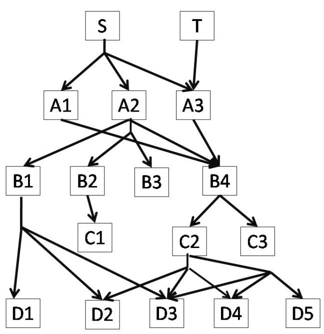

Fig. 2. A multi-standard directed hypergraph showing the breakdown of standards S and T.

For all what follow, we'll denote $\Im$ by the set of the top level standards in the multi-standard system.

B. A mathematical model of the graph structure of the SDR multi-standard system

A theoretical representation of the graph structure presented in subsection III-A can be concluded. Formally speaking, the graph structure of a multi-standard system can be viewed as a directed hypergraph defined by the couple (V, E), where the set of vertices V includes the blocks (functions and operators) present in the graph structure (example $V = \{S, T, A1, A2, A3, B1, \cdots, D4, D5\}$ in the graph structure of Fig. 2) and a directed hyperedge $e \in E$ will include the parent node as a tail node while all the necessary descendent node(s) capable of performing its task will form the head node(s) of e. So this means that whenever we have an "AND" dependency, the hyperarc is formed such that the parent node is the tail node and all the descendent nodes via this "AND" dependency are its head nodes. Whereas when faced with an "OR" dependency, the hyperarc will have the parent node as the tail node and only one of its descendent nodes (if more than one exists) via the corresponding "OR" dependency will be the head node. In this way, we form the set of hyperarcs E including all the "OR" and "AND" dependencies present in the corresponding graph structure. For instance, we have $(\{A2\}, \{B1\})$ and $(\{A2\}, \{B2, B3\})$ belong to the set of hyperarcs E of the directed hypergraph of Fig. 2.

C. A representation of one option of implementation

The graph structure of the multi-standard system supplies us with all the alternatives that can implement the design. In this part, we are going to explain how any one of these options of implementation can be illustrated. Note, however, that a certain selected option is characterized by the chosen common operators to install inside the design.

An illustration of a specific realized option of implementation will be defined by a **generated graph**, which provides a pictorial view of this option. A **generated graph** is a directed hypergraph obtained from the graph structure of a multi-standard system (introduced in III-A), by plotting the operators chosen to install in the design in such a way that they have empty forward stars, along with all the operators that they build, step by step, until they reach the functionalities of the top level standards.

Fig. 3 shows the generated graphs (obtained from Fig. 2) of two different options of implementation capable of realizing S and T. The first option is to choose the operators $D2, D3, D4, C1, \&B3$, whose generated graph is illustrated on the left part of Fig. 3. As for the second option, we choose to install each of $D2, D3, D4, C1, B3\&B4$ in the design which is also a feasible choice to implement the multi-standard system of Fig. 2, and whose generated graph is the one pictured on the right side of Fig. 3. Remark that there is a small difference between these two options. In the first, blocks $D2, D3, \&D4$ are used to realize the functionalities of both $A1\&A3$ passing through the $C2\&B4$ blocks. As for the second option, $D2, D3, \&D4$ are used to realize the $A3$ block (again passing by $C2\&B4$) while $B4$ is chosen to implement the functionalities of $A1$. The second case option represents an alternative in which certain lower level blocks are installed in the design, together with higher level ones which can be built by these of lower level (as $B4$ is installed in the design along with $D2, D3, \&D4$ which themselves can realize the functionalities of $B4$). Since this is a possible feasible option of implementation, so it was worth being considered.

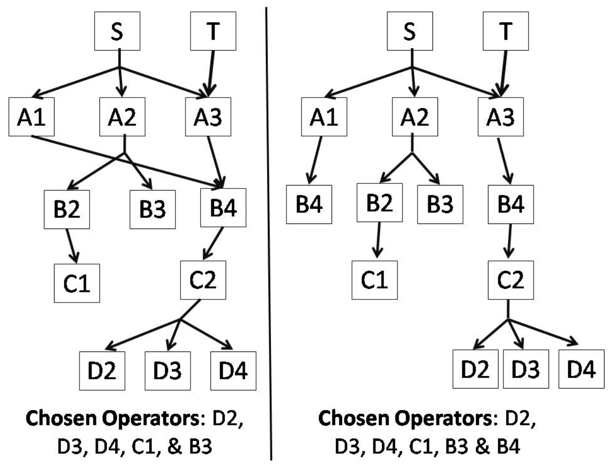

Fig. 3. The generated graphs (obtained from Fig. 2) of two different options of implementation

However, notice that the generated graph of the first option is a G-path with root $\Im$ while that of the second is not, as it's not a sub-hypergraph (the vertex block $B4$ plot twice). Remark that options resembling to the second choice always have a duplicated part, which contradicts to the illustration of a directed hypergraph and thus don't correspond to G-paths.

Consequently, the options of implementation can be split into those whose generated graphs are G-paths with root $\Im$, and those in which duplication occurs. However, it has been proved in [10] that the options which admit a duplication part can't in any way correspond to minimum cost designs. Thus we can restrict our choices to those of the form of G-paths with root $\Im$ when searching for a design with minimum cost, where the cost is calculated via the cost function presented in the following section. This idea will be exploited in the proposed algorithm presented in section V.

IV. The cost function

Recall that the graph structure of the multi-standard system provides all the options capable of implementing the standards to be supported. In this section, we will present a cost function which calculates the cost of any one of these options of implementation chosen to implement the multi-standard system.

Some necessary cost parameters were considered in the cost function introduced in [4]. A Building Cost (BC) and a Computational Cost (CC) were associated on each PE of the system, where a BC stands for the cost of the building PE capable of realizing a certain function and is just paid once independently of the number of times in which this PE is going to be called, while a CC is considered to be the time taken by a PE to compute a function and is paid every time it is invoked. Moreover, a parameter called the Number of Calls (NoCs) was associated to every BF-reduction of each hyperarc, corresponding to the necessary number of times in which children nodes at lower levels will be called by their parent node to perform the parent's node functionality.

Note that for what follows, the term weight will be used to represent the NoCs associated to any BF-reduction of a hyperarc in the directed hypergraph. This means that the number of times block x calls block y (which will be a number attached on the BF-reduction $(\{x\},\{y\})$ of a certain hyperarc e) will be abbreviated by $w_e(x,y)$.

An alternative theoretical formal expression of the cost function in [4] was provided in [11], which exploits various definitions and notations concerning directed hypergraphs. The cost function was written as:

$$CF = \sum_{y/BS(y)=\phi} \Big(\sum_{x/FS(x)=\phi} \sum_{P\ yx-path} CC(x) \times w(P)\Big) + \sum_{x/FS(x)=\phi} BC(x). \quad (2)$$

where:

- $\sum_{x/FS(x)=\phi} BC(x)$ represents the total sum of BCs of the blocks x such that $FS(x)=\phi$, which stand for the installed blocks in the design.
- $\sum_{P\ yx-path} CC(x) \times w(P)$ is the necessary computational cost imposed by the installed block x responsible for realizing the standard y (y is a highest level block because $BS(y)=\phi$ so it's a standard).

- $\sum_{x/FS(x)=\phi} \sum_{P\ yx-path} CC(x) \times w(P)$ stands for the total computational cost imposed by all the installed PEs x in the design to perform the functionality of the standard y.
- $\sum_{y/BS(y)=\phi} (\sum_{x/FS(x)=\phi} \sum_{P\ yx-path} CC(x) \times w(P))$ represents the total computational cost paid for all the standards.

The calculation process of the above cost function is as follows: select a top level standard y and one installed block x, then search for all paths P (in the G-path associated to the choice of implementation) from y to x (i.e yx-paths) in order to multiply the weight of each such paths by the CC of the installed block x. Repeat the same operation for each y standard and x installed block. In this way, we get the total CC of the system. Finally, we have to add the BC of each installed block.

V. A Minimum Cost Design Algorithm

In this section, we will present an algorithm which is capable of solving our optimization problem, stated as follows: given a directed hypergraph H representing all the alternatives capable of implementing a multi-standard design, find the option of implementation which yields the minimum cost (based on the cost function of equation 2).
As mentioned in III-C, we can restrict our search to the options which are illustrated as G-paths when seeking minimum cost designs.

Let H be a directed hypergraph representing the break-down of a multi-standard SDR system. Recall that $\Im \subseteq V(H)$ denotes the set containing the highest level blocks in H. We will define H_r, the directed hypergraph obtained from H, as follows:

- $V(H_r) = V(H) \cup \{r\}$.
- $E(H_r) = E(H) \cup \{E_r\}$ where $E_r = (\{r\}, \Im)$.

In fact, H_r is obtained by adding an imaginary top level vertex r to $V(H)$ and the hyperarc E_r to $E(H)$, where $\{r\}$ is the tail set of E_r and $\Im$ is that of its head. Since the vertex r plays the role of an imaginary highest level standard, so this changes the graph structure of the multi-standard system H, which possibly contains several top level blocks, into H_r which only contains one.

The parameters assigned to the entities of H_r will be:

- $CC(v)\ in\ H_r = CC(v)\ in\ H\ \forall\ v \in V(H)$
- $BC(v)\ in\ H_r = BC(v)\ in\ H\ \forall\ v \in V(H)$
- $w_{E_r}(r, v) = 1\ \forall\ v \in \Im$
- $w(x, y)\ in\ H_r = w(x, y)$ in H where $x, y \in V(H)$

Remark that no BC or CC is assigned to the vertex r due to its unnecessity.

In our algorithm when calculating the cost of an option, we will need to search for all the paths from each of the standards in $\Im$ to all the installed blocks in a selected option. This will be equivalent to searching for all the paths from only the vertex r in H_r to the same installed blocks in the option, thus reducing some steps of the algorithm. Besides, note that the weights on all the BF-reductions of the E_r hyperarc are set to 1. It is so in order to ensure that a path from r to an installed block will have the same weight as that of a path from a certain vertex in $\Im$ to the same installed block, and thus the cost of the design will not be influenced by the addition of the new vertex r.

Our algorithm is called the Minimum Cost Design (MCD) algorithm. The only input that it needs will be the directed hypergraph H_r obtained from H (where H represents the graph structure of a multi-standard system), together with the entities of H_r. We will also need to input the level of each block in H_r.

This algorithm will find all G-paths of H_r (representing certain options of implementation) in a step by step manner, generating options from others. It will compute the cost of every selected option to be tested (using the cost function of equation 2) and compare its cost to the previously examined G-paths by the algorithm. Finally, it will exhibit as an output the G-path with the minimum cost together with its corresponding cost found.

During the iterations of the algorithm we will introduce, for each G-path X selected, a vector k_v associated to every vertex $v \in V(X)$ defined recursively from the highest level nodes in X to the lowest, where the dimension of this vector will be derived as follows:

$$\begin{cases} dimk_r = 1; \\ dimk_v = \sum_{E \in BS_X(v)} \sum_{w \in T(E)} dim\ k_w\ \ \forall v \neq r; \end{cases} \tag{3}$$

Note that r will be the top level vertex of every G-path found.

notation: $k_v = (k_{1v}, k_{2v}, k_{3v}, \cdots, k_{(dim\ k_v)v})$ where each component of k_v will correspond to the weight of a path from r to v, and dimk_v will represent the number of such paths.

Furthermore in the algorithm, we will introduce a set Q in which the vertices of the option X in hand will be invoked bit by bit. However, a vertex to select at each step from Q will be that occupying the highest level among those present in Q which was imposed by the recursive definition of the vectors k_v for vertices v occupying high levels to lower level ones. Thus, the algorithm selects the element u in Q at every step such that: $l(u) = max\{l(w); w \in Q\}$.
Many variables have been introduced in the algorithm. We will explain the benefit of some of them.

- The variable R_P is introduced to occupy the total cost of the p^{th} G-path, selected from the set of options M.
- S is the variable in which we accumulate the cost of a certain G-path.
- A is the set which will contain all vertices v in the selected G-path X satisfying that $|FS_X(v)| = 0$
- SMin is an integer variable which will include the least cost of a G-path obtained so far.
- K is the variable in which we reserve the G-path with the least cost among those tested so far.

Here are the complete steps of the "MCD" algorithm:

```
Procedure(H_r, CC(v), BC(v), NoC(v))
begin
M = {({r}, φ)},  R_p := 0,  p := 1,  D := φ;
repeat
    select and remove X ∈ M
    if X = ({r}, φ)                                  |
      go to step U                                   | 1
    end-if                                           |
    S := 0, A := φ ;
    k_r := k_1r := 1      dim k_r := 1;              |
    for each v ∈ V(X)\{r} do                         |
      k_v = 0 vector,    dimk_v := 0;                | 2
    end-for                                          |
    Q = {r};
    repeat
    select and remove v ∈ Q
    for each E ∈ FS_X(v) do                          |
    begin                                            |
      for each h ∈ H(E) do                           | 3
      begin                                          |
        Q := Q ∪ {h}                                 |
        i := 1                                       |
        repeat                                       |
        if k_ih ≠ 0                                  |
         i := i + 1                                  | 4
        end-if                                       |
        until k_ih = 0                               |
       if v ≠ r                                      |
         for each E ∈ BS_X(v)                        |
           for each w ∈ T(E)                         |
             dim k_v := dim k_v + dim k_w            | 5
           end-for                                   |
         end-for                                     |
        end-if                                       |
        j := 0                                       |
        repeat                                       |
        k_(i+j)h = k_(j+1)v × w_E(v, h)              | 6
         j := j + 1                                  |
        until j := dim k_v                           |
        if |FS_X(h)| = 0                             |
         l := i                                      |
         repeat                                      |
           S := S + CC(h) × k_lh                     | 7
           l := l + 1                                |
         until l = i + dim k_v                       |
         A := A ∪ {h}                                |
        end-if                                       |
      end-for
    end-for
until Q = φ
repeat                                               |
select and remove v ∈ A                              | 8
  S := S + BC(v)                                     |
until A = φ                                          |
R_p := S
if p = 1                                             |
  SMin := R_p                                        | 9
  K := X                                             |
end-if                                               |
p := p + 1                                           | 10
R_p := 0                                             |
if p > 2                                             |
  if R_(p-1) < SMin                                  |
    SMin := R_(p-1)                                  | 11
    K := X                                           |
  end-if                                             |
end-if                                               |
```

STEP U
for each $u \in V(X)$ **/** $FS_X(u) = \phi$ **do**
begin
  for each $E \in FS_{H_r}(u)$ **do**
    $M := M \cup \{X + E\}$
  end-for
end-for **12**
until $M = \phi$
end-procedure.

Each of the 12 statements shown in the algorithm has its own significance. We'll explain the role of some of them.

1) Step 1 ensures that if $X = (\{r\},\phi)$ is the selected G-path from M, then we have to skip the calculation of the cost of X and go to Step U. This is because this G-path X has no technical significance for the implementation of the multi-standard SDR system, as r is an imaginary vertex added to H.
2) In step 2, as an initialization for the calculation of the cost of X, we fix $k_r = (1)$ vector and $k_v = 0$ vector $\forall\ v \neq r$. Note that k_v is a vector with unknown dimension at this step.
3) step 3 (easy).
4) The role of step 4 is to identify the index i of the first zero component k_{ih} of vector k_h so that we can update it after, rather than updating on previously settled numbers.
5) Step 5 is the one in which we compute dim k_v, as it's required for the next steps. This computation follows the equations in 3. Remark that if $v = r$, then there will be no need to compute the dimension because dim k_r is initialized to 1 in step 2.
   Note that since in this step we will need to calculate the dimension of the selected vertex v from Q, then all the entries of k_v should be already calculated and this is what imposed choosing the highest level block from Q every time we want to select a vertex. In face, if we select a vertex v from Q which doesn't occupy the highest level, it might be that its vector k_v has not been fully completed, as for example a higher level unselected vertex t from Q could impose new entries in k_v (if $\exists\ E \in FS_X(t)$ and $v \in H(E)$).
6) The sixth step consists in multiplying all the components of k_v by $w_E(v,h)$ in order to gain the weight of all the paths from r to h passing through v via the hyperarc E. However, we'll fill these new components of k_h starting from k_{ih}.
7) Step 7 is accessed only if $|FS_X(h)| = 0$ (i.e if h is an installed block) obeying the calculation of the cost formulated in equation 2. In such cases, we multiply the newly calculated components of k_h in step 6 by $CC(h)$ and add each one of them in the variable S. Moreover, the vertex h is added to the set A since in this case, h is a vertex with an empty forward star in X.
8) After having calculated the total computational cost S of the G-path X, we still have to add the Building Costs of all the installed blocks (which are accumulated in the set A)into S. This is achieved in step 8.
9) Step 9 consists in just putting the cost "$R1$" of the first selected option X in SMin with its corresponding G-path X in K, as this will be the first and only option tested so far and thus will represent the option with the least cost at this point.
10) The tenth step is responsible of initializing the next R_P to zero, in which we will be associating later the cost of the following G-path chosen from M.
11) Step 11 is responsible for updating SMin to the possible lower cost found (if it was less) and to update K to the corresponding lower cost G-path.
12) The final step generates G-paths of H_r from the G-path X. It searches for all vertices u which have empty forward stars in X and for each, find the hyperarcs $E \in FS_{H_r}(u)$ in order to add the generated option $X + E$ to M. It can be easily concluded that such generated options are also G-paths.

The MCD algorithm clearly doesn't seem to be a fast algorithm due to the large number of instructions and loops involved in its statements. Unfortunately, we weren't able to arrive to a simple form of the running time or memory space requirements of this algorithm. Consequently, our idea is to examine its performance by using a proposed program code of its statements. It might be a good tool, to run the program on various input examples in order to draw conclusions on the algorithm's complexity. This will be addressed in future work.

VI. Conclusion

Our research dwells with the optimization of the SDR multi-standard system using graph theory. This will lead to the construction of an optimal design. In this paper, we adopted various definitions and notations of directed hypergraphs to exhibit a new algorithm whose role is to identify the option of implementation of a multi-standard system with the minimum cost. This MCD algorithm provides the best optimal solution for our optimization problem. It searches for all the necessary options of implementation (missing some of the proved inefficient options for a minimum cost design in previous work) in a step by step manner and compares their costs in order to finally choose the option with the minimum cost.

Future work will include providing a program code idea of such an algorithm, whose input is a directed hypergraph and which needs to generate all the G-paths and calculate the cost of each. This will help establish the performance of this algorithm based on key parameters of the graph structure of a multi-standard system (number of levels, number of vertices, $\cdots$).

REFERENCES

[1] F.Jondral, "Parametrization: A technique for SDR implementation," Chapter 8 of Software Defined Radio Enabling Technologies, edited by W. Tuttlebee, Wiley, 2002.

[2] Alaus L, Palicot J, Roland C, Louet Y and Noguet D, Promising Technique of Parameterization For Recongurable Radio, the Common Operators Technique : Fundamentals and Examples, march 2009, Revue Journal of Signal Processing Systems, Springer New York.

[3] Sufi T. Gul: Optimization of Multi-Standards Software Defined Radio Equipments : A Common Operators Approach, PhD thesis, Rennes University (SUPELEC) 2009.

[4] Christophe Moy, Jacques Palicot, Virgillo Rodriguez, Denis Giri, "Optimal Determination of Common Operators for Multi-Standard Software-Defined Radio", 4th Karlsruhe Workshop on Software Radios, March 2006.

[5] Reinhard Diestel. Graph Theory. Fourth Edition 2010. Springer-Verlag, Heidelberg Graduate Texts in Mathematics, Volume 173.

[6] Claude Berge: Graphs and Hypergraphs, North-Holland, Amsterdam, 1973

[7] G. Gallo, G. Longo, S. Pallottino, and Sang Nguyen. Directed hypergraphs and applications. Discrete applied mathematics, 1993.

[8] Bruce Hajek, "A tutorial survey of theory and applications of simulated annealing," 24th IEEE Conference on Decision and Control, Dec 1985, pp.755-760.

[9] D.E.Goldberg, Genetic Algorithms in Search, *Optimization and Machine Learning* Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[10] Patricia Kaiser, Amine El Sahili, Yves Louet "Optimization of the SDR Multi-Standard system using Graph Theory", Journal of Interconnection Networks, submitted in 2011.

[11] Patricia Kaiser, Yves Louet, Amine El Sahili, "A cost function expression for SDR multi-standard systems design using directed hypergraphs", URSI General Assembly and Scientific Symposium , Istanbul, Turkey, August, 2011.