Logic Programming approaches for routing fault-free and maximally-parallel Wavelength Routed Optical Networks on Chip (Application paper)

MARCO GAVANELLI, MADDALENA NONATO
Department of Engineering, Ferrara University, Ferrara, Italy
(*e-mail:* `[marco.gavanelli,maddalena.nonato]@unife.it`)

ANDREA PEANO
QuanTek, Bologna, Italy
(*e-mail:* `andrea.peano@quantek.it`)

DAVIDE BERTOZZI
Department of Engineering, Ferrara University, Ferrara, Italy
(*e-mail:* `davide.bertozzi@unife.it`)

submitted 1 January 2003; revised 1 January 2003; accepted 1 January 2003

Abstract

One promising trend in digital system integration consists of boosting on-chip communication performance by means of silicon photonics, thus materializing the so-called Optical Networks-on-Chip (ONoCs). Among them, wavelength routing can be used to route a signal to destination by univocally associating a routing path to the wavelength of the optical carrier. Such wavelengths should be chosen so to minimize interferences among optical channels and to avoid routing faults. As a result, physical parameter selection of such networks requires the solution of complex constrained optimization problems. In previous work, published in the proceedings of the International Conference on Computer-Aided Design, we proposed and solved the problem of computing the maximum parallelism obtainable in the communication between any two endpoints while avoiding misrouting of optical signals. The underlying technology, only quickly mentioned in that paper, is Answer Set Programming (ASP). In this work, we detail the ASP approach we used to solve such problem.

Another important design issue is to select the wavelengths of optical carriers such that they are spread across the available spectrum, in order to reduce the likelihood that, due to imperfections in the manufacturing process, unintended routing faults arise. We show how to address such problem in Constraint Logic Programming on Finite Domains (CLP(FD)).

KEYWORDS: Answer Set Programming, Logic Programming Applications, Optical Networks on Chip, Constrained Optimization, Constraint Logic Programming on Finite Domains.

1 Introduction

Since photons move faster than electrons in the matter, and they dissipate lower power in the process, the new technology of silicon photonics is a great promise for small-scale ICT. It promises to provide unmatched communication bandwidth and reduced latencies with low energy-per-bit overhead. In recent years, remarkable advances of CMOS-compatible silicon photonic components have made it possible to conceive optical links and switching fabrics for performance- and power- efficient communication on the silicon chip. One proposal is to have silicon photonics-enabled on-chip interconnection networks implemented entirely in optics and using all-to-all conflict-free communication (leveraging the principle of wavelength-selective routing).

Wavelength-routed optical networks univocally associate the wavelength of an optical signal with a specific lightpath across the optical transport medium. They started to gain momentum in the domain of wide-area networks when it became clear that the electronics inside the optical network nodes were becoming the data transmission bottleneck (Berthold et al. 2008). Consequently, lightpaths in wavelength-routed networks were used to provide all-optical transmission between the source and the destination nodes (Chlamtac et al. 1992). This way, no optical-to-electrical-to-optical conversion and data processing were required at any intermediate node.

The recent advances of silicon photonics have raised a strong interest in using optical networks for on-chip communication (Optical Networks on Chip (ONoCs)). In this context, wavelength routing has been proposed as a way of relieving the latency and power overhead of electrically-assisted ONoCs to resolve optical contention. In fact, Wavelength-Routed Optical Networks on Chip (WRONoCs) are appealing as all-optical solutions for on-chip communication, since they avoid any form of routing and arbitration through the selection of disjoint carrier wavelengths for initiator-target pairs (Brière et al. 2007; Koohi et al. 2011; Tan et al. 2012).

Switching fabrics in a wavelength-routed ONoC are generally implemented with microring resonators (Bogaerts et al. 2012). These devices have a periodic transmittance characteristic, which means that they end up on resonance not only with one optical signal, but also with all those signals (if used) that are modulated on carrier wavelengths that are also resonant wavelengths of the microrings.

This issue raises a misrouting problem: one optical signal (or a significant fraction of its power) heading to a specific destination may end up being coupled onto another optical path, leading to a different destination. However, this problem has not been consistently addressed so far in ONoC literature, since the emphasis has been mainly on making the case for on-chip optical communication. As a result, wavelength-routed ONoC topologies are typically not refined with implementation details, but rather assessed by means of high-level power macromodels. The ultimate implication is that physical parameters such as microring resonator radii and carrier wavelengths are not selected, but simply addressed by means of symbolic assignments. Hence, the misrouting concern (in this paper explicitly addressed as a routing fault) is left in the background.

The unmistakable evidence of this trend is given by the fact that whenever research teams come up with actual photonic integrated circuits of wavelength-routed structures, the misrouting concern arises. For instance, in (Kaźmierczak et al. 2009) a 4×4 optical crossbar using wavelength routing is fabricated and tested. Since designers did not give too much importance to parameter selection during the design phase, they ended up choosing resonant peaks for their microring resonators that were not properly spaced throughout the available bandwidth. As a result, when injecting optical power on specific lightpaths, they detected significant power on unintended output ports of the device as well (an effect named *optical crosstalk*). Once deployed in a real system, their refined implementation may result in a misrouting fault and/or in error-prone communications, from the functional viewpoint. Kaźmierczak et al. (2009) consider this as a future optimization step of their work. Our research aims at bridging exactly this existing gap in wavelength-routed ONoC literature.

In previous work (Peano et al. 2016), we discussed the electronics and photonics design issues linked to the maximization of the parallelism in WRONoCs. As explained in that paper, the optimal design was found using Answer Set Programming (ASP), a technology still not very

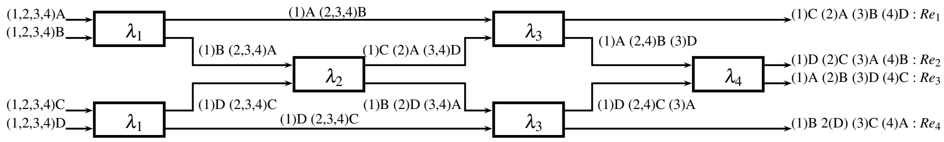

Fig. 1. A WRONoC topology connecting 4 senders (named A, B, C and D) to 4 receivers (Re_1 to Re_4) and using 4 carrier wavelengths (named λ_1 to λ_4). Numbers refer to communication channels, e.g., $(1,2)$A means the communication channels consisting of λ_1 and λ_2 originating from sender A.

well known in that research area. In this work, we take for granted the electronics and photonics issues and focus on the computational issues related to this hard optimization problem. We detail the ASP program used to solve the problem, and experimentally compare its performance with a Mixed-Integer Linear Programming (MILP) model. Another related problem was solved in (Nonato et al. 2017) through a MILP formulation. In this paper, we address the same problem in another logic language, namely Constraint Logic Programming on Finite Domains (CLP(FD)), and show that the CLP(FD) formulation is competitive with MILP and that it is easier to modify.

In the next section, we describe the two problems addressed in this paper. After some preliminaries (Section 3), we formalize the problem of maximizing the parallelism in a WRONoC (Section 4), then we describe the ASP program that solves such problem (Section 5). We then motivate the second problem, namely the uniform spreading of the selected resonances, and propose a CLP(FD) solution (Section 6). We show through experimental results (Section 7) that the proposed logic programming approaches have good performance with respect to mathematical programming formulations, and, finally, we conclude.

2 Problem description

In WRONoCs, n_R senders communicate with n_R receivers; each source-destination pair is associated with an optical channel using a specific wavelength for the optical carrier: the information originating from one sender is routed toward the correct receiver depending on the used carrier wavelength. In the same way, each receiver is able to receive communications from each of the n_R senders, distinguishing the correct sender through the wavelength of the carrier. For simplicity, instead of *wavelength of the carrier* we will often use just *wavelength* or *carrier*. Sender Se_1 uses disjoint wavelengths λ_1 to λ_{n_R} to communicate with receivers Re_1 to Re_{n_R}, respectively; at the same time, receiver Re_1 receives optical packets from senders Se_1 to Se_{n_R} on different wavelengths λ_1 to λ_{n_R}. More in general:

- each sender uses different wavelengths to communicate with the different receivers;
- each receiver receives information from different senders using different wavelengths.

Instead of using a new set of wavelengths, sender Se_2 reuses the same wavelengths used by Se_1.

The communication flows of a WRONoC topology can thus be abstracted by means of a Latin Square, that is a matrix $n_R \times n_R$ containing n_R values such that each row and each column contains n_R values. Each matrix value indicates the wavelength of the optical carrier that implements the communication between a specific sender-receiver pair.

The routing is done through optical devices called Photonic Switches (PSs); typical PSs have two input and two output ports and have a base resonance wavelength. They consist of two micro-rings, and the base resonance wavelength depends on the radius of the rings. If the signal in the first input port resonates with the PS, then it is deviated toward the first output; otherwise it is

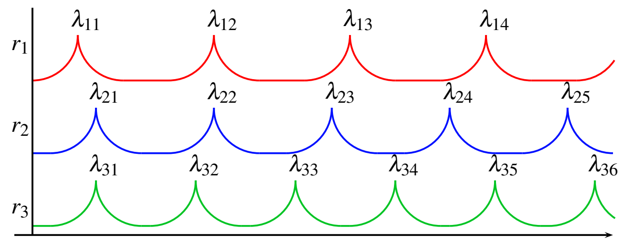

Fig. 2. An example of available spectrum, with three available radii $\mathscr{R} = \{r_1, r_2, r_3\}$ and a set of resonating wavelengths for each radius

passed to the second output port. The second input is treated symmetrically. A number of such devices build up a WRONoC, and various topologies have been proposed to ensure the correct routing of the information. Figure 1 shows one of such topologies, connecting four senders (A, B, C, and D) to four receivers (Re_1 to Re_4), and using four wavelengths ($\lambda_1, \dots, \lambda_4$). For example, if sender A uses wavelength λ_1, the signal resonates with the first PS and exits from the first output port; here it is sent to a PS that resonates with λ_3 and is sent to its second output port. It then enters the first input port of a PS resonating with λ_4 and is then sent to its second output port. Note that each of the four receivers can distinguish the origin of the information through the used wavelength; e.g., when Re_3 senses a signal of wavelength λ_1, the sender must have been A.

It can be observed that each PS can resonate not only to its base wavelength, but also to a number of other harmonic wavelengths; Table 1 is an example of a very small instance showing the set $\Lambda_r = \{\lambda_{r,1}, \lambda_{r,2}, \dots\}$ of resonance wavelengths for different values of radii. This effect can be exploited to increase the communication parallelism, as a sender-receiver pair could communicate not only through the base wavelength but also using some of the harmonics. In such a case, the communication channel between two endpoints consists of two or more carriers, with different wavelengths, resonances of the same radius. However, it might be the case that the i-th harmonic of one PS could be equal (or too close) to the j-th harmonic of another one: in such a case the laser beam would be incorrectly deviated in the WRONoC topology, and a so-called *misrouting* or *routing fault* would occur.

In Figure 2 three possible radius values are available $\mathscr{R} = \{r_1, r_2, r_3\}$; for each radius r_i, there is a set of resonating wavelengths $\{r_{i,1}, r_{i,2}, \dots\}$ that can be selected as carriers. Suppose that $n_R = 2$; this means that 2 radii must be selected (out of the 3 available). Note that $\lambda_{21} = \lambda_{31}$; this means that if both r_2 and r_3 are selected, wavelength λ_{21} cannot be selected as carrier, because it would be incorrectly routed, since it also resonates with radius r_3. The same holds also for $\lambda_{12} = \lambda_{22}$. Also, the wavelength λ_{14} is very close to λ_{35}; in real settings there exist always imprecisions in the fabrication process, so it is not advisable to select wavelengths that are too close: a minimum distance $\Delta\lambda$ should separate any two selected wavelengths.

One possible solution would be to select r_1 and r_2; in such a case, three wavelengths can be selected for each radius without routing faults: in fact for r_1 the set of wavelengths $\{\lambda_{11}, \lambda_{13}, \lambda_{14}\}$ can be selected, while for r_2 any three wavelengths can be selected out of the four that do not conflict with r_1: λ_{21}, λ_{23}, λ_{24}, and λ_{25}. The obtained parallelism is 3.

It is then important to select n_R different radii, taken from the set of available radii $\mathscr{R}$, and for each selected radius r select n_λ resonating wavelengths (taken from the set Λ_r of harmonics of the radius r) such that each sender-receiver pair can use n_λ wavelengths (harmonics of the same radius) while avoiding routing faults; the objective is maximizing the number n_λ. This problem

Table 1. $\mathscr{T}_{\mathscr{R}}$ *with radii varying from 5 to* 8μm

r	R_r	$\lvert\{\lambda_{r,j}\}\rvert$	$\lambda_{r,1}$ [nm]	$\lambda_{r,2}$ [nm]	$\lambda_{r,3}$ [nm]	$\lambda_{r,4}$ [nm]	$\lambda_{r,5}$ [nm]	$\lambda_{r,6}$ [nm]	$\lambda_{r,7}$ [nm]
1	5μm	5	1496.4	1521.3	1547.1	1573.8	1601.4		
2	6μm	6	1500.5	1521.3	1542.7	1564.8	1587.5	1610.8	
3	7μm	6	1503.4	1521.3	1539.6	1558.4	1577.7	1597.4	
4	8μm	7	1505.6	1521.3	1537.3	1553.7	1570.4	1587.5	1604.9

was solved in (Peano et al. 2016) through an ASP formulation, that was only cited in that paper. In this paper, instead, we detail the ASP program in Section 5.

After finding the maximum parallelism n_λ obtainable, one has to choose a suitable solution amongst the (possibly, many) solutions providing the same optimal value of parallelism. In (Nonato et al. 2017), it was found that the wavelengths found when solving the first problem could be unevenly spread in the available spectrum. This introduced a second problem: given n_R and n_λ, find n_R radii values and $n_R \times n_\lambda$ wavelengths (n_λ per radius) such that the selected wavelengths are as evenly spread as possible. Such problem was solved in (Nonato et al. 2017) with a MILP formulation. In this work, we address the same problem in another logic programming language, namely CLP(FD). We show that the CLP(FD) program is competitive in terms of performance with the MILP approach. Moreover, we found that a different formulation is more adherent to the WRONoC design problem, and that the CLP(FD) program can be easily modified to account for the revised formulation. The MILP approach, instead, must be subject to major rewriting in order to tackle this revised formulation.

The complete solution process consists of two phases: in the first, the maximum obtainable parallelism is obtained through an ASP program. The optimal value of parallelism is then provided to the second phase: a CLP(FD) program that, given a target value of parallelism, computes a set of wavelengths that 1) achieve the given parallelism level and 2) are as equally spaced as possible in the available spectrum.

3 Preliminaries

3.1 Answer Set Programming

Answer Set Programming (ASP) is a class of logic programming languages that rely on the stable model semantics (Gelfond and Lifschitz 1988), also known as answer set semantics. We assume a basic familiarity with logic programming and its syntax; for an introduction the reader can refer to (Lloyd 1987). A logic program consists of a set of rules a :- $l_1, l_2, \ldots, l_n$ where a is an atom (also called the *head* of the rule), and the l_i are literals (called the *body* of the rule).

Literals and rules containing no variables are called *ground*. We denote as $gr(r)$ all possible instantiations of the rule r of the program Π, on the basis of ground facts of the program. The *ground istantiation* of Π consists of all ground instances of rules in Π, i.e., $gr(\Pi) = \bigcup_{r\in\Pi} gr(r)$. For any set M of atoms from Π, let Π_M be the program obtained from Π by deleting (i) each rule that has a negative literal $\neg B$ in its body with $B \in M$ and (ii) all negative literals in the bodies of the remaining rules. Since Π_M is negation free, it has a unique minimal Herbrand model. If this model coincides with M, then M is a Stable Model of Π (Gelfond and Lifschitz 1988).

Among the dialects of ASP, we use the language of the grounder Gringo (Gebser et al. 2009), that extends the basic logic programming syntax with a number of features.
Counting (Simons et al. 2002). If $a_1, a_2, a_3, \ldots$ are atoms, and l and u are integers, the aggregate $l\{a_1, a_2, a_3, \ldots\}u$ is true for every set S of atoms including from l to u members of $\{a_1, a_2, a_3, \ldots\}$, i.e., $l \leq |\{a_i \in S\}| \leq u$. Trivial bounds can be omitted.
Summation. If $a_1, a_2, a_3, \ldots$ are atoms and $v_1, v_2, v_3, \ldots$ are integers, the aggregate $l\sharp\mathbf{sum}[a_1 = v_1, a_2 = v_2, a_3 = v_3, \ldots]u$ is true for every set S of atoms such that the sum of v_i over included members a_i of $\{a_1, a_2, a_3, \ldots\}$ is in the interval $[l, u]$: $l \leq \sum_{i:a_i \in S} v_i \leq u$.

Usually, ASP solvers (Simons et al. 2002; Lin and Zhao 2004; Giunchiglia et al. 2006; Leone et al. 2006; Gebser et al. 2011) work in two stages. In the first, called *grounding*, the program is converted into an equivalent ground program. The second stage is devoted to looking for stable models (answer sets) of the ground program.

3.2 Constraint Logic Programming on Finite Domains

Constraint Logic Programming (CLP) is a class of logic programming languages (Jaffar and Maher 1994) that extends Prolog with the notion of *constraints*. Each language of the CLP class is identified with a *sort*; one of the most popular is CLP(FD), on the sort of Finite Domains. CLP(FD) is particularly suited to solve Constraint Satisfaction Problems (CSPs). A CSP consists of a set of decision variables, each ranging on a finite domain, and subject to a set of relations called *constraints*. A solution to the CSP is an assignment of values taken from the domains to the respective variables, such that all the constraints are satisfied.

A Constraint Optimization Problem (COP) is a CSP with an additional objective function, that must be maximized or minimized. A solution of a COP is optimal if it satisfies all the constraints and, amongst the solutions of the CSP, it maximizes (or minimizes) the objective function.

4 Maximizing parallelism

We now give a formalization of the problem of finding the maximum parallelism. A set $\mathscr{R}$ of possible radius values is given. For each $r \in \mathscr{R}$, a set $\Lambda_r = \{\lambda_{rj}\}$ of resonance wavelengths is also given. Two wavelengths λ_{ri}, λ_{sj} are in conflict if $|\lambda_{ri} - \lambda_{sj}| \leq \Delta\lambda$ for a given $\Delta\lambda \geq 0$.

The core decisions concern which resonances should be selected for each radius. To model this decision we use the boolean variable $x_{rj} \in \{0,1\}$ to state whether the resonance wavelength λ_{rj} is selected for radius r. The problem can be formalized as the following COP:

$$P(s) = \max : \min_{r \in 1..|\mathscr{R}|} \{q_r \mid q_r > 0\} \text{ s.t.} \tag{1}$$

$$q_r = \sum_{\lambda_{rj} \in \Lambda_r} x_{rj} \qquad \forall\, r \in 1..|\mathscr{R}| \tag{2}$$

$$s_r = \begin{cases} 0 & q_r = 0 \\ 1 & q_r > 0 \end{cases} \qquad \forall\, r \in 1..|\mathscr{R}| \tag{3}$$

$$\sum_{r \in 1..|\mathscr{R}|} s_r = n_R \tag{4}$$

$$x_{rj} = 1 \Rightarrow s_{r'} = 0 \qquad \begin{array}{r} \forall r, r' \in 1..|\mathscr{R}| \forall j \in 1..|\Lambda_r| \\ \text{s.t. } \exists i \in 1..|\Lambda_{r'}| \wedge |\lambda_{rj} - \lambda_{r'i}| \leq \Delta\lambda \end{array} \tag{5}$$

q_r represents the number of selected resonances for radius r. The objective function (1) maximizes the parallelism in the selected radius with the least parallelism, since the global network parallelism is bounded by the channel with lowest parallelism. In practice, we maximize the minimum parallelism that can be sustained by all of the wavelength channels. Constraints (2) define the number q_r of selected elements in row r. Constraints (3) define whether the radius r is selected ($s_r = 1$) or not ($s_r = 0$). Constraint (4) imposes to select exactly n_R radii. Finally, Constraints (5) prevent routing faults; they are imposed for each λ_{rj} and $r' \neq r$ such that λ_{rj} is conflicting with some resonance $\lambda_{r'i}$ in radius r'.

Consider, for example, the instance in Table 1, suppose that $n_R = 3$, i.e., three radii must be selected, and $\Delta\lambda = 0$, i.e., two wavelengths are in conflict only if they are identical. One solution is to select radii 2, 3, and 4, i.e., $s_1 = 0$ and $s_2 = s_3 = s_4 = 1$ (satisfying constraint 4). Notice that in Table 1 $\lambda_{r,2} = 1521.3$ for all values of r. From constraint 5, selecting such wavelength for some radius (e.g., for radius 2, i.e. $x_{22} = 1$) means that all other radii must not be selected: contradiction. Thus clearly $x_{r,2} = 0$ for all radii r. Also, $\lambda_{2,5} = \lambda_{4,6}$, so by constraint (5), they cannot be selected, since both radii 2 and 4 are selected. All other wavelengths can be selected; i.e. $x_{2,1} = x_{3,1} = x_{4,1} = x_{3,3} = \cdots = x_{4,7} = 1$ is a possible assignment. We have that $q_1 = 0$, $q_2 = 4$, $q_3 = 5$ and $q_4 = 5$. The minimum of the not-null q_i is $q_2 = 4$, that is also the value of the objective function for this assignment.

5 An ASP program to compute maximum WRONoC parallelism

The ASP program takes as input an instance provided with facts

lambda(R, L_{min}, $L_{nominal}$, L_{max})

expressing the fact that the radius R resonates at the wavelength $L_{nominal}$; due to variations in temperature and other uncertainties, the actual wavelength might change, with a maximum deviation $\Delta\lambda$, i.e., in the range $[L_{min}, L_{max}] = [L_{nominal} - \Delta\lambda, L_{nominal} + \Delta\lambda]$.

Predicate `radius/1` is true for the available radii (the elements of the set $\mathscr{R}$), while `lambda/2` is true for the available wavelengths for each radius (elements of the set Λ_R):

```
lambda(R,L) :- lambda(R,_,L,_).
radius(R) :- lambda(R,_).
```

From the set of available wavelengths, some are chosen as transmission carriers. Predicate sL(r, j) is true if the wavelength j is chosen for the radius r, i.e., iff $x_{r,j} = 1$ in the COP of Eq. (1-5):

```
{ sL(R,L) : lambda(R,L) }.
```

The set of chosen radii is then given by:

```
sR(R) :- sL(R,_).
```

sR(r) is true iff $s_r = 1$ in the COP of Eq. (1-5). The number of chosen radii must be equal to the number n_R of devices that need to communicate:

:- not $n_R \leq$ { sR(R): radius(R) } $\leq n_R$. (6)

In order to avoid routing faults (constraint (5)), we define a conflict relation. Two wavelengths $L1$ and $L2$ are in conflict if they are selected for different radii and the intervals $[L^1_{min}, L^1_{max}]$ and $[L^2_{min}, L^2_{max}]$ have non-empty intersection.

```
conflict(R1,R2,L1,L2):- lambda(R1,L1_min,L1,L1_max), R1!=R2,
                        lambda(R2,L2_min,L2,L2_max), L1 < L2, L1_max ≥ L2_min.
conflict(R1,R2,L1,L2):- lambda(R1,L1_min,L1,L1_max), R1!=R2,
                        lambda(R2,L2_min,L2,L2_max), L1>L2, L2_max ≥ L1_min.
conflict(R1,R2,L,L):- lambda(R1,L1_min,L,L1_max), R1!=R2,
                      lambda(R2,L2_min,L,L2_max),
```

Also, it might be the case that two wavelengths for the same radius are in conflict

```
conflict(R,L1,L2):- lambda(R,L1_min,L1,L1_max),
          lambda(R,L2_min,L2,L2_max), L1<L2, L1_max>=L2_min.
```

Note that the `conflict` predicate depends only on the input data, and not on the wavelengths that must be chosen as carriers. The truth of the `conflict` atoms in the answer set is decided in the grounding phase, and does not require a search during the computation of the answer set.

If wavelength $L1$ of radius $R1$ is in conflict with some wavelength of radius $R2$, then $L1$ and $R2$ cannot be both selected; if two wavelengths are in conflict within the same radius, they cannot be selected:

```
:-conflict(R1,R2,L1,L2),radius(R1),radius(R2),sL(R1,L1),sR(R2).
:-conflict(R,L1,L2), sL(R,L1), sL(R,L2), L1<L2.
```

Finally, the objective is to maximize the number of wavelengths selected for each radius. Predicate `countR`/2 provides the number of selected resonances for each radius, and corresponds to constraint (2):

```
countR(R,Qr) :- radius(R), Qr>0, Qr=#count{ 1,L : sL(R,L) }.
```

Predicate `bp`/1 provides the minimum number of resonances that have been selected varying the radius; the objective is maximizing such value, as in Eq (1):

```
#maximize{ P : bp(P) }.
```

Predicate `bp`/1 could be implemented following the definition (Eq. 1), i.e.:

```
bp(P):- P=#min{ Qr:countR(R,Qr) }, P>0.
```

however, a more efficient version is using chaining and an auxiliary predicate:[1]

```
auxbp(N):- countR(_,N).
auxbp(N+1) :- auxbp(N), N < F, maxF(F).
bp(P):- auxbp(P), not auxbp(P-1).
```

where `maxF` computes the maximum number of wavelengths that might be selected, and that can be calculated during grounding.

6 Spacing the selected resonances

As will be shown in the experimental results (Section 7), the ASP program in Section 5 was very efficient in computing the maximum parallelism. On the other hand, after analyzing the provided solutions, it was found that often the selected wavelengths were unevenly spread in the available

[1] We thank one of the anonymous reviewers for suggesting this improved formulation.

spectrum. Since, due to imprecisions in the fabrication process, the actual wavelengths might be different from the computed ones, it might be the case that two selected wavelengths become too close in the actual device, and the two wavelengths might be confused raising a routing fault. As often done in the electronic component industry, after fabrication each device is checked, and if it is not working properly it is discarded.

A second-level optimization could then be performed in order to select, amongst the possibly many resonances that provide the same optimal parallelism, those ones that are more evenly spread in the available spectrum, with the idea that maximizing the distance between selected wavelengths can reduce the likelihood that the actual wavelengths are too close, and, consequently, that the device has to be discarded.

The ASP program in Section 5 was then modified to take as input the parallelism to be achieved, and to have as objective to uniformly spread the selected resonances. The performances, however, were not satisfactory, and a complex MILP model, based on network flow, was devised (Nonato et al. 2017). Another logic programming based approach was developed in CLP(FD); we describe it in next section.

6.1 A CLP(FD) approach to the problem of spacing selected resonances

As already said, in this second optimization phase, we have as input a value n_λ of parallelism to be achieved. The objective is to select n_R values of radii and $n_R \times n_\lambda$ resonance wavelengths (n_λ for each radius) such that the selected wavelengths are as equally spread in the available spectrum as possible.

In the CLP program, we focused on modeling the problem with fewer variables than the MILP and ASP formulations. In MILP the problem is modeled with one variable for each pair (r, λ) stating that resonance λ is selected for radius r. Similarly, in ASP there is predicate $\texttt{sL}(r, \lambda)$ that is true if λ is selected for radius r. In the CLP program, we have n_R variables $R_1, \dots, R_{n_R}$ that range over the set $\mathscr{R}$ of possible radii; each of the R_i represents one chosen value of radius.

A common rule of thumb to have efficient CLP(FD) programs is to employ the so-called *global constraints* (Régin 1994), i.e., constraints that involve a large number of variables and for which powerful propagation algorithms have been designed in the past. The idea is that using global constraints, the propagation can exploit more global information (opposed to the local information used in arc-consistency propagation and its variants) because each constraint has visibility of many variables at the same time.

Clearly, all the radii must be different, so we have

$$\mathit{alldifferent}([R_1, \dots, R_{n_R}])$$

where *alldifferent* (Régin 1994) imposes that all variables take different values.

The selected resonances are represented through an $n_R \times n_\lambda$ matrix M; each element M_{ij} ranges over the set of available wavelengths, and represents the j-th wavelength selected for radius R_i.

Each of the variables in the i-th row of matrix M is linked to the radius variable R_i; for each i, M_{ij} should be a resonance wavelength of radius R_i. This can be imposed through a `table` constraint (Zhou 2009). The `table` constraint is useful to define new constraints by listing the set of available tuples; in our case it lists the set of pairs (R, L) for which R is a radius and L one of its corresponding resonance wavelengths.

Constraint Logic Programming is particularly effective at solving scheduling problems, mainly due to the effectiveness of the *cumulative* constraint. The *cumulative* constraint considers a set

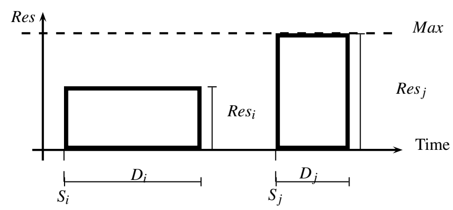

Fig. 3. Example of cumulative constraint with two tasks

of tasks, each described with a start time, a duration and a resource consumption, and it ensures that in each time the sum of the resources consumed by the scheduled tasks does not exceed a given limit *Max*. Let *S* be the list of start times, *D* that of durations and *Res* that of resource consumptions,

$$cumulative(S,D,Res,Max)$$

is true if (see Figure 3)

$$\forall t \qquad \sum_{i:S_i \le t \le S_i + D_i} Res_i \le Max.$$

The three lists *S*, *D* and *Res* can contain variables with domains or constant values, and the constraint removes, through constraint propagation, inconsistent values. In the particular case in which $\forall i, Res_i = 1$ and $Max = 1$, the *cumulative* constraint imposes that the tasks should not overlap in time.

In the problem of maximally spreading wavelengths, we model the selected wavelengths as tasks of a scheduling problem. Each of the M_{ij} elements of the M matrix is considered as the start time of a task, with a total of $n_R \times n_\lambda$ tasks. All the tasks have the same duration: one variable *Dist* models the duration of all the tasks. If we now impose[2]

$$cumulative([M_{ij} | i \in 1..n_R, j \in 1..n_\lambda], [Dist]_{n_R}, [1]_{n_R}, 1) \tag{7}$$

this constraint imposes that all wavelengths do not overlap, and that they are spaced of at least *Dist* units. The objective will be to find the maximum possible value for variable *Dist* that does not cause any conflict.

To model conflicts between resonances, we recall that each resonance wavelength for a chosen radius R_k must be different from all the wavelengths M_{ij} selected for another radius R_i. We first explain how to model the relation between a radius R_i and the set of resonance wavelengths, then we provide a set of global constraints to model conflicting wavelengths.

The relation between a radius R_r and the corresponding resonances $\Lambda_{R_r} = \{\lambda_{r,1}, \lambda_{r,2}, \dots\}$ is imposed through an `element` constraint (Van Hentenryck and Carillon 1988). The $element(I,L,X)$ constraint ensures that the I-th element of the list L has value X. We represent the set Λ_r as a list of constrained variables $[\lambda_{r,1}, \lambda_{r,2}, \dots]$; the length of the list is the number of resonances in the radius with the maximum number of resonances $Max_{\sharp\lambda} = \max_k |\Lambda_k|$. The i-th element of the list, $\lambda_{r,i}$, is subject to the constraint

$$element(R_r, \mathscr{T}^i_{\mathscr{R}}, \lambda_{r,i})$$

where $\mathscr{T}^i_{\mathscr{R}}$ is the i-th column of Table 1. To account for the different number of resonances in different radii, the list is filled with dummy values.

[2] where we indicate with $[X]_n$ the list $[X,X,X,...]$ containing n times element X.

Since the list of resonance wavelengths consists of different wavelengths, in order to model conflicts between the selected resonances for one radius and the resonance wavelengths for other radii one might impose

$$\forall i \in 1..n_R, \forall k \in 1..n_R, i \neq k, \quad \mathit{alldifferent}([M_{ij} | j \in 1..n_\lambda] \cup \Lambda_{R_k}) \tag{8}$$

that are $n_R(n_R - 1)$ *alldifferent* constraints, each containing $n_\lambda + Max_{\sharp\lambda}$ variables. However, one might notice as well that all the elements in the M matrix are different, meaning that instead of (8) one can impose

$$\forall k \in 1..n_R, \quad \mathit{alldifferent}([M_{ij} | i \in 1..n_R, i \neq k, j \in 1..n_\lambda] \cup \Lambda_{R_k}) \tag{9}$$

that are n_R constraints each containing $(n_R - 1)n_\lambda + Max_{\sharp\lambda}$ variables.

Please, note the symbols: each radius r has a number of resonance wavelengths, the j-th is named λ_{rj}. Out of the λ_{rj}, some are selected as carriers: the i-th wavelength selected for radius r is named M_{ri}.

Finally, the objective is maximizing variable *Dist*, that is a lower bound to the minimal distance between selected wavelengths.

6.1.1 Breaking Symmetries

The problem contains a number of symmetries:

- The order in which the resonance wavelengths appear in one of the rows of the matrix L is not important: given a solution, another solution can be obtained by swapping two elements. More importantly, swapping two elements in an assignment that is not a solution, provides another non-solution.
- Swapping the order of two radii (both in the list of radii and as rows of the M matrix) provides an equivalent solution.

Removing symmetries is considered important to speedup the search. We tried several strategies, and the best was the following:

- the rows of the M matrix are sorted in ascending order. This could be done imposing $M_{ij} < M_{i,j+1}$, but since all wavelengths must be at least *Dist* units apart, the following constraint gives stronger propagation:

$$\forall i \in 1..n_R, \forall j \in 1..n_\lambda - 1 \quad M_{ij} + Dist \leq M_{i,j+1}$$

- the first column of the matrix is sorted in ascending order:

$$\forall i \in 1..n_R - 1 \quad M_{i,1} + Dist \leq M_{i+1,1}$$

6.1.2 Objective function

As previously said, the objective is to maximize the value assigned to variable *Dist*, that represents the minimum distance between two selected resonances.

Adding known bounds of the objective function can strengthen the propagation. Clearly, the maximum possible value for *Dist* is obtained if all the selected wavelengths are equally spaced. As $n_R n_\lambda$ resonances are selected, the following bound holds:

$$(n_R n_\lambda - 1)Dist \leq \left(\max_{i \in 1..n_R, j \in 1..n_\lambda} M_{i,j} \right) - \left(\min_{i \in 1..n_R, j \in 1..n_\lambda} M_{i,j} \right).$$

Given the symmetry breaking constraints, $\min_{i,j} M_{i,j} = M_{1,1}$, while $\max_{i,j} M_{i,j}$ is the maximum of the last column of the M matrix: $\max_i M_{i,n_R}$.

6.2 A refined CLP(FD) approach

As will be shown in the experimental results (Section 7), the CLP approach just shown did not reach the performance of the MILP program in (Nonato et al. 2017). However, a closer look to the set of selected wavelengths (both in the MILP and in the CLP approaches) showed that a further refinement of the problem formulation was necessary. In fact, in order to minimize the likelihood of routing faults, a selected resonance $M_{r,i}$ should be as far as possible not only from the other selected resonances $M_{s,j}$, but also from all the resonance wavelengths of the selected radii ($\lambda_{R,i}$ for all the selected R and all i), independently from the fact that they are also selected as carriers or not.

Considering this effect, the MILP model in (Nonato et al. 2017) can no longer be used and a major rewriting is required, because the problem can no longer be modeled as a constrained shortest path.

The CLP program, instead, can be easily modified to account for this effect. A first tentative would be to consider also the (non-selected) resonance wavelengths of selected radii as tasks. The *alldifferent* Constraints in Eq. (9) can be rewritten as *cumulative* constraints, in which the tasks corresponding to selected wavelengths have duration $Dist$, while those corresponding to non-selected wavelengths have a very short duration (value 1nm is suitable in our instances):

$$\forall k \in 1..n_R,\ cumulative([M_{ij} | i \in 1..n_R, i \neq k, j \in 1..n_\lambda] ++ \Lambda_k,\\ [Dist]_{(n_R-1)n_\lambda} ++ [1]_{Max_{\sharp\lambda}},\qquad [1]_{(n_R-1)n_\lambda + Max_{\sharp\lambda}}, 1)$$

where the symbol ++ stands for list concatenation. However, with this approach each selected resonance would be at least *Dist* units from the *following* resonance (either selected or non-selected), but no constraint prevents it to be very close to the *preceding* non-selected resonance.

A possible solution would be to represent selected resonances M_{ij} as tasks with start time $M_{ij} - \frac{Dist}{2}$ and duration $Dist$, i.e., M_{ij} would be the center of the task instead of its start time. This modification introduces a large overhead, due to the fact that the constraint associated with the summation operator propagates very poorly.

A more effective CLP(FD) modeling is to introduce a duration *Dist* also for non-selected resonances (of selected radii). However, this would introduce a minimal distance also between two non-selected wavelengths, a constraint which is not required for WRONoCs, and would lead to sub-optimal solutions. We decided to use the *resource* parameter of the *cumulative* constraint to avoid the collision between tasks of non-selected resonances. Each of the $Max_{\sharp\lambda}$ non-selected resonances is modelled as a task of duration *Dist* and using 1 resource unit (see Figure 4). The limit of resources is exactly $Max_{\sharp\lambda}$, so that tasks of non-selected resonances can overlap. Each selected resonance is modeled as a task of duration *Dist* and using all resources ($Max_{\sharp\lambda}$):

$$\forall k \in 1..n_R,\ cumulative([M_{ij} | i \in 1..n_R, i \neq k, j \in 1..n_\lambda] ++ \Lambda_k,\\ [Dist]_{(n_R-1)n_\lambda + Max_{\sharp\lambda}}, [Max_{\sharp\lambda}]_{(n_R-1)n_\lambda} ++ [1]_{Max_{\sharp\lambda}}, Max_{\sharp\lambda})$$

In this way a task corresponding to a selected resonance cannot overlap neither with tasks of selected resonances, nor with those of non-selected resonances, and must be at least at *Dist* distance from any other resonance of selected radii.

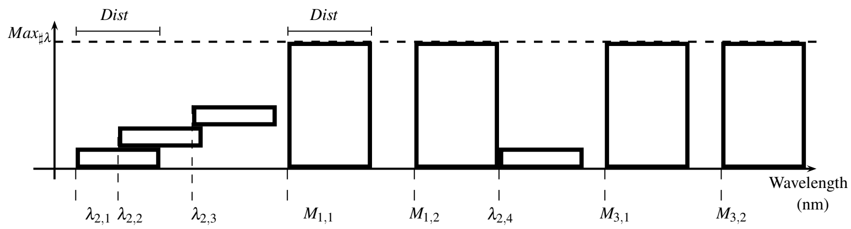

Fig. 4. Example of *cumulative* for spacing the selected resonances ($M_{i,j}$) at a minimum distance *Dist*. Non-selected resonances ($\lambda_{i,j}$) can be close to each other, but they cannot be close to selected resonances.

7 Experimental results

In the experimental campaign in (Peano et al. 2016), the focus was computing the maximum obtainable parallelism varying the fabrication parameters, including the possible deviations of the laser wavelengths and the radius imprecisions during fabrication of the device. In this work, instead, we report the timing results of the ASP formulation and of a MILP model.

We considered a set of radii ranging from 5nm to 30nm in steps of 0.25nm; this yields 104 possible radii. In order to compute the corresponding resonance wavelengths, an Electromagnetic Model (Parini et al. 2011) computes the transmission responses; with the selected values of radii, 1850 resonances are obtained, with a number of resonances per radius ranging from 5 to 28.

We compare the ASP program described in Section 5 with a MILP model that is a linearization (with standard techniques) of the problem defined in Section 4. The employed ASP solver is clasp 4.5.4, and the MILP solver is Gurobi 7.0.1; Gurobi was run through its Python interface. All experiments were run on a computer with an Intel Core i7-3720QM CPU running at 2.60GHz, with 16GB RAM, using Linux Mint 18.1 64-bit. All experiments were performed using only one core. All the code and instances are available on the web.[3]

The results are plotted in Figure 5 for ideal lasers (left), and for $\Delta\lambda = 1$nm (right). The ASP program has usually better performances than the MILP model, and in particular in the non-ideal case, in which finding an assignment satisfying all constraints is more difficult, while Gurobi seems more efficient in the case with less tight constraints, in which the difficulty is more driven by the need to find an optimal solution.

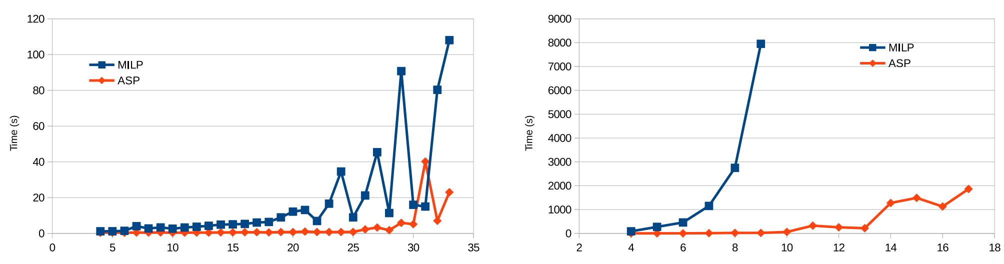

Fig. 5. Comparison of MILP and ASP running time when maximizing the minimum parallelism with $\Delta\lambda = 0$ (left) and $\Delta\lambda = 1$nm (right)

[3] http://www.ing.unife.it/en/research/research-1/information-technology/computer-science/artificial-intelligence-group/research-projects/wronoc/

7.1 Maximally spreading resonances

The second set of experiments assesses the performance of Logic Programming approaches in the problem of selecting carrier wavelengths maximally spread on the available spectrum. We compare the performance of the CLP(FD) program described in Section 6.1 with the MILP flow model in the same instances considered in (Nonato et al. 2017) and for which Gurobi did not run out of memory.

$n_R \times n_\lambda$	MILP	CLP(FD)	refined CLP(FD)
4×1	508.68	14.45	10.41
8×1	563.47	24.81	393.14
4×4	2973.77	Time Out	2303.85

Table 2. *Comparison of MILP-Gurobi and CLP(FD)-ECL^{i}PS^e run time on the problem of maximizing the distance between wavelengths. MILP and CLP(FD) maximize the distance only between wavelengths selected as carriers, while the* refined CLP(FD) *model finds selected wavelengths at the maximum distance to any resonance wavelength of selected radii.*

The experiments were run on the same computer given earlier, using Gurobi 7.0.1 as MILP solver and ECL^{i}PS^e 6.1 (Schimpf and Shen 2012) as CLP(FD) solver. The time-out was 3600s. While the MILP approach is very effective in the largest instance, the CLP(FD) program is more effective in the small instances. On the other hand, the refined CLP(FD) program, that models more closely the requirements of the WRONoC architecture, is better than the MILP approach in all instances. Note also that the MILP program (Nonato et al. 2017) is very tailored toward solving the problem of maximally spreading the selected resonances and has to be completely rewritten for modifications of the problem, such as adding further constraints or changing the objective function. The Logic Programming approaches, instead, are more general and modifiable, as can be seen from the relatively small modifications required to extend the first CLP approach (Section 6.1) to the refined CLP program (Section 6.2).

It is also worth noting that Gurobi is a commercial program, while both clingo and ECL^{i}PS^e are developed as open-source programs.

8 Conclusions

We presented two problems arising in the industry of opto-electronic components, in particular in Wavelength-Routed Optical Network on Chip (WRONoC) design. The first problem, published in (Peano et al. 2016) arose because in the electronic research the maximal communication parallelism obtainable with a WRONoC was unknown. The problem was solved with an ASP program, that was mentioned, but not described in detail, in (Peano et al. 2016). We described the ASP program and compared experimentally its performance with a Mixed-Integer Linear Programming (MILP) approach.

The second problem (Nonato et al. 2017) comes from the observation that, once the maximum parallelism level is found, it is also of interest to design the WRONoC in the safest way, despite small variations that might occur in the fabrication process. In order to maximize the probability that the device is able to function correctly, the selected wavelengths used as carriers have to be as far as possible one from the other. Such problem was approached in (Nonato et al. 2017) through a MILP formulation. In this work, we presented a Constraint Logic Programming on Finite Domains (CLP(FD)) program, showed that it has performances competitive with the MILP approach and found that it is easier to modify it to take into consideration further aspects in the WRONoC design.

In both cases, Logic Programming approaches have proven to be competitive with mathematical programming technologies, and that Logic Programming has promising techniques to address problems in the new area of Wavelength-Routed Optical Network on Chip (WRONoC) design.

In future work, we plan to address the two described problems combining the best features of CLP and ASP; a number of Constraint Answer Set Programming solvers have been proposed and are natural candidates for this research direction (Mellarkod et al. 2008; Wittocx et al. 2008; Drescher and Walsh 2010; Janhunen et al. 2011; Balduccini and Lierler 2012; Liu et al. 2012; Bartholomew and Lee 2014; Susman and Lierler 2016).

References

BALDUCCINI, M. AND LIERLER, Y. 2012. Practical and methodological aspects of the use of cutting-edge ASP tools. In *Practical Aspects of Declarative Languages - 14th International Symposium, PADL 2012, Philadelphia, PA, USA, January 23-24, 2012. Proceedings*, C. V. Russo and N. Zhou, Eds. Lecture Notes in Computer Science, vol. 7149. Springer, 78–92.

BARTHOLOMEW, M. AND LEE, J. 2014. System aspmt2smt: Computing ASPMT theories by SMT solvers. In *Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings*, E. Fermé and J. Leite, Eds. Lecture Notes in Computer Science, vol. 8761. Springer, 529–542.

BERTHOLD, J., SALEH, A. A. M., BLAIR, L., AND SIMMONS, J. M. 2008. Optical networking: Past, present, and future. *Journal of Lightwave Technology 26,* 9 (May), 1104–1118.

BOGAERTS, W., DE HEYN, P., VAN VAERENBERGH, T., DE VOS, K., SELVARAJA, S. K., CLAES, T., DUMON, P., BIENSTMAN, P., VAN THOURHOUT, D., AND BAETS, R. 2012. Silicon microring resonators. *Laser Photonics Rev. 6,* 1, 47–73.

BRIÈRE, M., GIRODIAS, B., BOUCHEBABA, Y., NICOLESCU, G., MIEYEVILLE, F., GAFFIOT, F., AND O'CONNOR, I. 2007. System level assessment of an optical NoC in an MPSoC platform. In *2007 Design, Automation Test in Europe Conference Exhibition*. IEEE, 1–6.

CHLAMTAC, I., GANZ, A., AND KARMI, G. 1992. Lightpath communications: an approach to high bandwidth optical WAN's. *IEEE Transactions on Communications 40,* 7 (Jul), 1171–1182.

DRESCHER, C. AND WALSH, T. 2010. A translational approach to constraint answer set solving. *TPLP 10,* 4-6, 465–480.

GEBSER, M., KAMINSKI, R., OSTROWSKI, M., SCHAUB, T., AND THIELE, S. 2009. On the input language of ASP grounder Gringo. In *LPNMR*, E. Erdem, F. Lin, and T. Schaub, Eds. LNCS, vol. 5753. Springer, 502–508.

GEBSER, M., KAUFMANN, B., KAMINSKI, R., OSTROWSKI, M., SCHAUB, T., AND SCHNEIDER, M. 2011. Potassco: The Potsdam Answer Set Solving Collection. *AI Communications 24,* 2, 107–124.

GELFOND, M. AND LIFSCHITZ, V. 1988. The stable model semantics for logic programming. In *Proceedings of International Logic Programming Conference and Symposium*, R. Kowalski, Bowen, and Kenneth, Eds. MIT Press, 1070–1080.

GIUNCHIGLIA, E., LIERLER, Y., AND MARATEA, M. 2006. Answer set programming based on propositional satisfiability. *J. Autom. Reasoning 36,* 4, 345–377.

JAFFAR, J. AND MAHER, M. J. 1994. Constraint logic programming: a survey. *J. Log. Program. 19/20*, 503–581.

JANHUNEN, T., LIU, G., AND NIEMELÄ, I. 2011. Tight integration of non-ground answer set programming and satisfiability modulo theories. In *Working notes of the 1st Workshop on Grounding and Transformations for Theories with Variables*. Vancouver, BC, Canada, 1–13.

KAŹMIERCZAK, A., BOGAERTS, W., DROUARD, E., DORTU, F., ROJO-ROMEO, P., GAFFIOT, F., VAN THOURHOUT, D., AND GIANNONE, D. 2009. Highly integrated optical 4×4 crossbar in silicon-on-insulator technology. *Journal of Lightwave Technology 27,* 16 (Aug), 3317–3323.

KOOHI, S., ABDOLLAHI, M., AND HESSABI, S. 2011. All-optical wavelength-routed NoC based on a novel hierarchical topology. In *Proceedings of the Fifth ACM/IEEE International Symposium*. 97–104.

LEONE, N., PFEIFER, G., FABER, W., EITER, T., GOTTLOB, G., PERRI, S., AND SCARCELLO, F. 2006. The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Log. 7,* 3, 499–562.

LIN, F. AND ZHAO, Y. 2004. ASSAT: computing answer sets of a logic program by SAT solvers. *Artif. Intell. 157,* 1-2, 115–137.

LIU, G., JANHUNEN, T., AND NIEMELÄ, I. 2012. Answer set programming via mixed integer programming. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*, G. Brewka, T. Eiter, and S. A. McIlraith, Eds. AAAI Press.

LLOYD, J. W. 1987. *Foundations of Logic Programming, 2nd Edition*. Springer.

MELLARKOD, V. S., GELFOND, M., AND ZHANG, Y. 2008. Integrating answer set programming and constraint logic programming. *Ann. Math. Artif. Intell. 53,* 1-4, 251–287.

NONATO, M., BERTOZZI, D., GAVANELLI, M., AND PEANO, A. 2017. A network model for routing-fault-free wavelength selection in WRONoCs design. In *International Network Optimization Conference 2017*. Electronic Notes in Discrete Mathematics.

PARINI, A., RAMINI, L., BELLANCA, G., AND BERTOZZI, D. 2011. Abstract modelling of switching elements for optical networks-on-chip with technology platform awareness. In *Proceedings of the Fifth International Workshop on Interconnection Network Architecture: On-Chip, Multi-Chip*. INA-OCMC '11. ACM, New York, NY, USA, 31–34.

PEANO, A., RAMINI, L., GAVANELLI, M., NONATO, M., AND BERTOZZI, D. 2016. Design technology for fault-free and maximally-parallel wavelength-routed optical networks-on-chip. In *Proceedings of ICCAD'16, the* 35^{th} *IEEE/ACM International Conference on Computer-Aided Design, Austin, Texas*. IEEE/ACM, 3:1–3:8. http://doi.acm.org/10.1145/2966986.2967023.

RÉGIN, J. 1994. A filtering algorithm for constraints of difference in CSPs. In *Proc. 12th National Conf. on Artificial Intelligence*, B. Hayes-Roth and R. E. Korf, Eds. AAAI Press / The MIT Press, 362–367.

SCHIMPF, J. AND SHEN, K. 2012. $\mathrm{Ecl}^{\mathrm{i}}\mathrm{ps}^{\mathrm{e}}$ - from LP to CLP. *TPLP 12,* 1-2, 127–156.

SIMONS, P., NIEMELÄ, I., AND SOININEN, T. 2002. Extending and implementing the stable model semantics. *Artif. Intell. 138,* 1-2 (June), 181–234.

SUSMAN, B. AND LIERLER, Y. 2016. SMT-based constraint answer set solver EZSMT (system description). In *Technical Communications of the 32nd International Conference on Logic Programming, ICLP 2016 TCs, October 16-21, 2016, New York City, USA*, M. Carro, A. King, N. Saeedloei, and M. De Vos, Eds. OASICS, vol. 52. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 1:1–1:15.

TAN, X., YANG, M., ZHANG, L., JIANG, Y., AND YANG, J. 2012. A generic optical router design for photonic network-on-chips. *Journal of Lightwave Technology 30,* 3 (Feb), 368–376.

VAN HENTENRYCK, P. AND CARILLON, J. 1988. Generality versus specificity: An experience with AI and OR techniques. In *Proc. 7th National Conf. on Artificial Intelligence*, H. E. Shrobe, T. M. Mitchell, and R. G. Smith, Eds. AAAI Press / The MIT Press, 660–664.

WITTOCX, J., MARIEN, M., AND DENECKER, M. 2008. The IDP system: a model expansion system for an extension of classical logic. In *Proceedings of Workshop on Logic and Search, Computation of Structures from Declarative Descriptions (LaSh)*. 153–165.

ZHOU, N.-F. 2009. Encoding table constraints in CLP(FD) based on pair-wise AC. In *Logic Programming, 25th International Conference, ICLP 2009, Pasadena, CA, USA, July 14-17, 2009. Proceedings*, P. M. Hill and D. S. Warren, Eds. Lecture Notes in Computer Science, vol. 5649. Springer, 402–416.