RADMAX: Risk And Deadline Aware Planning for Maximum Utility

Jingkai Chen, Cheng Fang, Christian Muise, Howard Shrobe, Brian C. Williams, and **Peng Yu**
Computer Science and Artificial Intelligence Laboratory, MIT
32 Vassar Street
Cambridge, Massachusetts 02139

Abstract

Current network approaches aim to maximize network utilization when routing flows. While such approaches are fast and usually result in acceptable behavior, existing methods are not *mission aware*. There is no concept of utility maximization, no capability to handle flows with specified deadlines and loss requirements, and no guarantees over the probability of network saturation.

In this paper, we present *RADMAX*: a system for Risk And Deadline Aware Planning for Maximum Utility based on constraint programming, which allows us to handle higher level mission specifications. We show the correctness of RADMAX with respect to loss and delay bounds, provide results for the optimality of RADMAX with respect to the mission utility, and review current results on computational performance.

Introduction

Research in network science has progressed at a rapid pace in terms of hardware and controllers. Given the advances in underlying infrastructure, the next step is to build network configuration planners which reason over higher level abstractions. Solving the network configuration problem with mission specifications would allow future networks to plan over mission specifications, such deadlines for file transfers, schedules for VOIP calls, and guarantees over loss and delay of flows.

Researchers in artificial intelligence have successfully deployed configuration planners for space systems, with successes including the Livingstone system for reconfiguration on the Deep Space One probe (Muscettola et al. 1998). This has motivated the development of *RADMAX*, a system for Risk And Deadline Aware Planning for Maximum Utility, a model-based network configuration manager.

Given a model of the current state of the underlying network in terms of bandwidth, loss, and delay on network links, and given mission spefications for flows, RADMAX plans routes, application of forward error correction (FEC) for loss reduction, and bandwidth allocations, with guarantees on the probability of success. Decision making is done by leveraging constraint programming encodings and techniques which have found success in analogous domains.

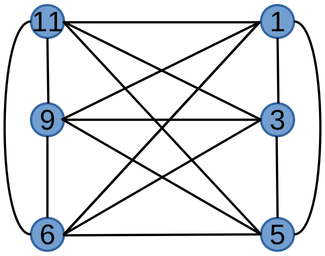

Figure 1: Network topology for the example problem.

The contributions of this work are as follows: 1) We present a model of network state and flow requirements which describe a network with mission specifications; 2) We provide a constraint programming encoding of the model; 3) We provide preliminary results regarding the utility of solutions, as well as first computational results for the scalability of the approach.

Problem Specification

In this section, we provide a formal definition of the network configuration problem with mission specifications. We begin with a motivating example problem, from which we will define the network configuration planning problem. We will then outline the inadequacies of a reactive based approach, and highlight the desired features of an automated network configuration planner.

Example 1. *(Replanning a FTP transfer with a deadline) We have the network topology as in Figure 1, with six nodes in a fully connected network. For each link, the bandwidth available is* $5Mbps$*, with loss 1.5%, and 10 millisecond delay.*

In specifying the mission, we use time index t*, such that the start time of the mission occurs at* $t = 0$*. Initially, there is only an FTP transfer from 9 to 3, of size* $560Mb$*. The transfer must be completed by time* $t = 120s$*, and has no delay or loss constraints. At time* $t = 30s$*, a video flow from 9 to 3 starts, which requires* $1Mbps$ *bandwidth, less than 1% loss, and up to 15 millisecond delay.*

We want to route both flows, such that the FTP transfer finishes by the set time limit, and such that the video flow *is placed on a route satisfying the bandwidth, loss and delay constraints. In addition to the choice of routes, we must also choose the configuration of each flow. The configuration choices included dropping the flow, applying FEC with 3 source packets and 1 parity packet, or routing without FEC.*

The example above serves to illustrate the most important features of a network configuration planning problem. In such specifications we must capture features of the network, the mission requirements for the flows, and the allowed configurations for the flows.

Formally, the network configuration problem specifications are defined as follows.

Definition 1. *(Network configuration problem specification) We consider the Network Configuration Problem to consist of the following network specifications:*

- *A set of nodes:* $N = \{1, ..., n\}$*;*
- $\forall i \in N\ j \in N$, $BW_e[i, j]$ *is the maximum bandwidth between two nodes;*
- $\forall i \in N\ j \in N$, $L_e[i, j]$ *is the expected loss between two nodes; and*
- $\forall i \in N\ j \in N$, $D_e[i, j]$ *is the expected delay between two nodes.*

The problem also contains the following flow requirement specifications:

- *A set of flows:* $M = \{1, ..., m\}$*;*
- $\forall k \in M$, $BW_f[k]$ *is the minimum required throughput of flow k;*
- $\forall k \in M$, $L_f[k]$ *is the maximum allowable loss of flow k;*
- $\forall k \in M$, $D_f[k]$ *is the maximum allowable delay of flow k;*
- $\forall k \in M$, $source[k] \in N$ *is the start node of flow k;*
- $\forall k \in M$, $sink[k] \in N$ *is the destination node of flow k; and*
- H_f *is the maximum allowable hops of flows.*

The problem also contains the following information on available configurations:

- *For each flow* $k \in M$*, a set of configurations:* $C_k = \{1, ..., c_k\}$*;*
- *Bandwidth on link function* $b : C_k \times \mathbb{R} \rightarrow \mathbb{R}$*, such that* $b(c_i, \tau)$ *is the actual bandwidth on link required after applying FEC* c_i *to a flow with required throughput* τ*;*
- *Loss on link function* $l : C_k \times [0, 1] \rightarrow [0, 1]$*, such that* $l(c_i, \delta)$ *gives the reduced loss resulting from adopting configuration* c_i *on a link with loss* δ*; and*
- *Utility* $u : C_k \rightarrow \mathbb{R}$*, such that* $u(c_i)$ *gives the utility of choosing configuration* c_i*.*

In describing the network features, each link from node i to node j ($i, j \in N$) has three characteristics: bandwidth $BW_e[i, j]$, loss $L_e[i, j]$, and delay $D_e[i, j]$. As the network is meshed, $BW_e[i, j]$ is set to 0 if nodes i and j are not directly connected.

For flow requirements, each flow $k \in M$ has start node at $source[k]$, and destination node $sink[k]$. In addition, the throughput allocated to flow has minimum required value $BW_f[k]$. Upper bounds on the allowed cumulative delay and loss along the assigned path path are also restricted to be below $L_f[k]$ and $D_f[k]$ respectively. We also model H_f an upper bound on the number of hops allowed for any flow.

Each flow k has a set of possible configurations C_k, which represent drop, normal, and various FEC settings. Each choice of configuration leads to different bandwidth on link, loss on link and utilities, calculated according to functions b, l and u respectively.

Current network approaches are reactive controls based and do not consider mission specifications. The behavior observed with current systems given the problem in Example 1 is as follows. At time $t = 0$, the FTP will be placed on the link $9 \rightarrow 3$, and use up all $5Mbps$ of the available bandwidth. At $t = 30s$, there would be approximately $150Mb$ of the FTP transfer completed. However, the controller will react to the new video transfer, and also place the video flow on the $9 \rightarrow 3$ link. As a result, the remaining $410Mb$ of the FTP transfer will only have $4Mbps$ bandwidth allocated. The FTP transfer will then require an additional 100 seconds to complete, and thus miss the transfer deadline.

By inspection, it can be confirmed that, by placing the FTP on an alternative route from 9 to 3, for example on links $9 \rightarrow 1$ and $1 \rightarrow 3$, we are able to meet the mission specifications for both flows, provided that we can also apply an FEC to correct for the link loss. We thus require a mission aware network configuration planner that autonomously produce such plans. In the subsequent section, we examine some relevant literature on techniques relevant to our solution method.

Related work

While we are not aware of mission level network configuration planning, there has been recent work in formulating network design as optimization problems. Typically, such works assume uncertain bandwidth demands for flows, and output the required capacities for links (Johnston, Lee, and Modiano 2013; Ben-Ameur and Kerivin 2005). Our problem is related in the sense that we have no control over the network capacities, but we may choose to actuate on the flow to exert control over bandwidth demands.

Network design algorithms typically make use of numerical optimization techniques. However, the network configuration planning problem has many similarities to the vehicle routing problem (VRP) (Kilby and Shaw 2006). The VRP is an extension of the traveling salesman problem, and attempts to find minimal length tours given a set of destinations to be visited by multiple vehicles. In addition, there are goods to be delivered at each destination, and vehicles have finite capacity for transporting goods. In both cases, we are required to find paths visiting a subset of nodes, while respecting cumulative constraints, in our case loss and delay bounds, in the VRP case capacity bounds for vehicles.

In the VRP community, current state of the art software based on constraint programming techniques are able to find solutions to benchmark problems with 50 destination nodes within a 1000 node graph in minutes (Hall 2016). We thus chose to base our encoding and solution methods, described in subsequent sections, on those found in the VRP community.

Constraint Modeling and Encoding

In this section, we describe the model used to encode the network configuration planning problem. We first provide a brief overview of the model for FEC used. We then describe the decision variables used to describe the choice of routes and actuation, as well as the auxiliary variables required for constraint checking. Lastly we describe the set of constraints which ensure that the routes and actuations are feasible given the network and meet the mission specifications.

FEC modeling

RADMAX was developed as part of the EdgeCT DARPA project, and thus must model well-defined actuators. Packet FEC is one such actuator, used to reduce loss on link at the expense of consuming more bandwidth.

When FEC is applied, q packets of source are sent with p packets of parity, such that whenever at least q packets are received out of the $p+q$ sent, then the source can be entirely recovered. In the case when less than q packets are received, then only $q' \leq q$ the number of received source packets can be recovered.

The effect on the bandwidth is thus straightforward: for throughput requirement BW, applying q source and p parity FEC means we have bandwidth over each link of

$$\frac{p+q}{p}BW$$

.

We approximated the effect of FEC on loss as follows. For each FEC block of q source and p parity packets, sent over link with loss δ, we may consider the expectation of the *proportion* of packets successfully sent. We will calculate

$$f_{fec}(q,k,1-\delta) = \frac{1}{q}E\left[\text{source packets recovered}\right]$$

For convenience, define the probability mass function of a binomial distribution as

$$B(q,q',1-\delta) = \binom{q}{q'}(1-\delta)^n\delta^{q-q'}$$

for q the number of trials, q' the number of successes, and $1-\delta$ the probability of success for each trial.

For FEC, we consider first the case when at least q packets were delivered:

$$\begin{aligned}
&\frac{1}{q}E\left[\text{source packets recovered when at least q delivered}\right]\\
=&\frac{1}{q}\sum_{i=q}^{p+q} h \times B(p+q,i,1-\delta)\\
=&\sum_{i=q}^{p+q} B(p+q,i,1-\delta)
\end{aligned} \tag{1}$$

Consider now the case when fewer than q packets were delivered:

$$\begin{aligned}
&\frac{1}{q}E\left[\text{source packets recovered when fewer than q delivered}\right]\\
=&\frac{1}{q}\sum_{i=1}^{q-1}\sum_{j=1}^{i} jP(\text{i delivered})P(\text{j of i source}|\text{i delivered})\\
=&\frac{1}{q}\sum_{i=1}^{h-1}\sum_{j=1}^{i} j \times B(p+q,i,\delta)B(i,j,\frac{q}{p+q})
\end{aligned} \tag{2}$$

Summing the two terms we find

$$\begin{aligned}
&f_{fec}(q,p,1-\delta)\\
&=\sum_{i=q}^{p+q} B(p+q,i,1-\delta)\\
&+\frac{1}{q}\sum_{i=1}^{q-1}\sum_{j=1}^{i} j \times B(p+q,i,1-\delta)B(i,j,\frac{q}{p+q})
\end{aligned} \tag{3}$$

Given that FEC with q source and p parity was applied, expected loss on a link with loss δ is thus

$$L_{FEC}(q,p,1-\delta) = 1 - f_{fec}(q,p,1-\delta)$$

Variables

In encoding the network configuration planning problem, we create the following variables:

- $\forall i \in N,\ k \in M,\ s[i,k] \in N$ is the direct successor of flow k on vertex i;
- $\forall i \in N, k \in M, l[i,k]$ is the cumulative loss of flow k on vertex i;
- $\forall i \in N,\ k \in M,\ d[i,k]$is the cumulative delay of flow k on vertex i;
- $\forall i \in N,\ k \in M,\ h[i,k]$is the cumulative hops of flow k on vertex i;
- $\forall k \in M, z[k] \in \{1, ..., c_k\}$ is the configuration of flow k;
- $\forall k \in M,\ c \in H_k,\ bw[k,c]$ is the minimum required throughput of flow k;
- $\forall k \in M, c \in H_k, i \in N, j \in N, l_e[c,i,j]$ is the expected loss between two vertices when a flow with configuration w passes this edge; and
- $\forall k \in M,\ c \in H_k,\ i \in N,\ j \in N,\ d_e[w,i,j]$ is the expected delay between two vertices when a flow with configuration w passes this edge.

As in encodings common to constraint programming of VRP (Kilby and Shaw 2006; Bent and Van Hentenryck 2004), we define a set of successor variables s. Intuitively, every successor variable denotes the direct successor node of a node A on a flow's routing path with an integer domain whose values represent the nodes connected to vertex A. Note that the successor variables on the same node for different flows are different. Thus, $s[i,k]$ for $i \in N, k \in M$ is the successor variable for node i and flow k. Given a set of assignments to successor variables of a flow, we may recursively extract the path as the set of successors from the source of the flow.

In addition to successor variables, for each node i and flow k, we also have variables l, d, and h, representing cumulative loss, delay and hops when flow k reaches vertex i. These variables are used to maintain upper bounds on the cumulative loss, delay and hops.

Lastly, we also create variables z which describe the choice of configurations. We create the auxiliary variables bw and l_e which describe the effects on bandwidth required on link and loss on link.

Constraints

In this section we describe the constraints used to represent the effects of configuration choices, constraints for network capacity, and constraints to enforce mission specifications.

We begin with constraints representing the effect of FEC. Assume that we have a mapping F_k which gives us a tuple (q,p) of source and parity packets, for any choice of configuration $z[k]$ for flow k:

- When flows dropped:

$$\begin{aligned} (z[k]=c) \wedge (F_k[c]=(0,0)) \Rightarrow \\ (bw[k,z[k]]=0) \\ \wedge\,(l_e[k,z[k],i,j]=0) \\ \wedge\,(d_e[k,z[k],i,j]=0) \\ \forall k \in M, i \in N, j \in N \end{aligned} \tag{4}$$

- FEC configuration for each flow not dropped:

$$\begin{aligned} (z[k]=c) \wedge (F_k[c] \neq (0,0)) \Rightarrow \\ \left(bw[k,z[k]] = \frac{p+q}{q} BW_f\right) \\ \wedge\,(l_e[k,z[k],i,j] = L_{FEC}(q,p,L_e[i,j])) \\ \wedge\,(d_e[k,z[k],i,j] = D_e[i,j]) \\ \forall k \in M, c \in C_k, i \in N, j \in N \end{aligned} \tag{5}$$

When the flow is dropped, then the cumulative loss and delays, as well as the bandwidth required on link is set to zero. This allows the loss and delay constraints to be trivially satisfied, and the flow does not take up space on the links. However, when the flow is not dropped, the bandwidth on link and loss are calculated given the model for FEC above. Note that application of FEC does not change the delay on link.

Routing of flows is encoded using a $ProperCircuit$ constraint, as in VRPs.

- ProperCircuit: Each flow circuit visits a subset of N

$$ProperCircuit(s[:,k]) \qquad \forall k \in M \tag{6}$$

- The successor of each flow's end should be its start:

$$s[sink[k],k] = source[k] \qquad \forall k \in M \tag{7}$$

$ProperCircuit$ (van Hoeve and Katriel 2006) is a global constraint commonly used in VRP that enforces the requirement for a set of nodes with one circuit visiting once a subset of the nodes. If a node is not connected to any other node, its successor is itself. For example. $\{s[1,1]=2, s[2,1]=3, s[3,1]=1, s[4,1]=4, s[5,1]=5, s[6,1]=6\}$ is a proper circuit, because $\{1,2,3\}$ are in a loop and 4,5,6 point to themselves. We also add a dummy link for each flow, such that the successor of its sink is the source. With this dummy link, the path for every flow is a cycle. An example is given in Figure 2.

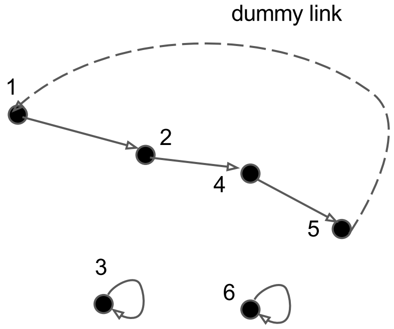

Figure 2: An example of successor assignments satisfying $ProperCircuit$, for a flow with source 1 and sink 5.

We must also ensure that the flows, given chosen configurations, are routed according to limits on link capacities.

- Edge bandwidth capacity constraint:

$$\sum_{k \in \{k \in M | (s[i,k]=j) \wedge (sink[k] \neq i)\}} bw[k,z[k]] \leq BW_e[i,j] \qquad \forall i \in N, j \in N \tag{8}$$

For each vertex i, the consistency check of bandwidth capacity is performed on all the links. For a link (i,j), if a flow k has passed this link such that $s[i,k]=j$, the throughput requirement of the flow k will considered unless the link (i,j) is a dummy path of flow k such that $sink[k]=i$. Lastly, because the successors of isolated vertices are themselves and these self-loops are also counted, the maximum bandwidth from each vertex to itself should be set as a positive infinite value to satisfy the bandwidth constraint.

Recalling that the flows have upper bounds over allowed accumulated loss, delay and number of hops, we define the following constraints.

- Loss constraints (conservative approximation with the union bound):

$$\begin{aligned} l[source[k],k] = 0 \\ l[s[i,k],k] = l[i,k] + l_e[z[k],i,s[i,k]] \\ l[sink[k],k] \leq L_f[k] \\ \forall k \in M, i \in \{i \in N | (i \neq s[i,k]) \wedge (i \neq sink[k])\} \end{aligned} \tag{9}$$

- Delay constraints:

$$\begin{aligned} d[source[k],k] &= 0 \\ d[s[i,k],k] &= d[i,k] + d_e[z[k],i,s[i,k]] \\ d[sink[k],k] &\leq delayf[k] \\ \forall k \in M, i \in \{i \in N | (i \neq s[i,k]) &\wedge (i \neq sink[k])\} \end{aligned} \tag{10}$$

- Hops Constraints:

$$\begin{aligned} h[source[k],k] &= 0 \\ h[s[i,k],k] &= h[i,k] + 1 \\ h[sink[k],k] &\leq H_f \\ \forall k \in M, i \in \{i \in N | (i \neq s[i,k]) &\wedge (i \neq sink[k])\} \end{aligned} \tag{11}$$

For each flow k, except for the isolated nodes, delay is accumulated from the source along the path, stopping when the flow arrives at the sink. We require that the accumulated delay at the sink is less than that allowed in the specifications. To account for the dummy link, we do not accumulate the delay from the sink to the source. Unlike bandwidth constraints, the loss of flow k is not coupled with other flows. As the isolated nodes are not connected to the sink directly or indirectly (because of proper circuit propagation), they do not influence the accumulated delay on every sink.

Similar encodings are used for loss and delay. Note that we chose to enforce the loss bound using by summing loss along the path. This is a conservative approximation using the Union Bound, which is true regardless of whether the losses are independent. The form of the constraint does not change if we assume independence: we can simply impose a constraint over the sum of the log loss.

Objective

In our encoding, the total utility is a linear sum of the utilities for each flow, for utility function for each flow p.

$$\max \sum_{k \in M} p(z[k], h[sink[k], k] \tag{12}$$

In our formulation, we would like to use the minimal amount of FEC such that all specifications are met. This allows us to have spare bandwidth for unexpected flows which may arrive during execution. Further, we would like to have short routes, so that we penalize the number of hops required to arrive at the sink. Lastly, we derive zero utility for any dropped flow.

Results

We tested our encoding using four scenarios, each intended to demonstrate a different feature. The constraint programs were solved using an in-house solver (OpSat-v3 2018), on a single 3.40GHz core with 32GB memory.

In each case, the network had topology as give in Figure 1. Each link had bandwidth 10Mbps, with loss 2% and delay 10 milliseconds. The configurations allowed for each flow included drop, normal actuation, and FEC settings (3,1), (5,2), (2,1), and (5,4).

Vignette 1: Rate control

In this vignette, we have a single FTP flow from 1 to 11, with 560Mb to be transferred over 180 seconds. The allowed loss is 3%. The example vignette was used to demonstrate correctness in the choice of FEC, as none will be necessary in this case.

Averaging over 10 runs, the problem was solved to optimality in 0.0364 seconds. The resulting solution placed the FTP flow on the direct link from 1 to 11 with no FEC applied, as expected.

Vignette 2: FEC application

In this vignette, we have two flows from 9 to 3. The first is an FTP flow with 900Mb to be transferred in 120 seconds, with allowed loss 3%. Additionally, there is a flow representing a VOIP call which requires 0.1Mbps bandwidth, but can only tolerate 0.5% loss.

Averaging over 10 runs, the problem was solved to optimality in 0.1988 seconds. The resulting solution placed both flows on the direct link. In addition, (3,1) FEC was applied on the VOIP call to reduce the loss to that required.

Vignette 3: Large number of flows

In this vignette, we have 30 flows. For each pair of nodes, we have a VOIP call flow call which requires 0.1Mbps bandwidth, but can only tolerate 0.5% loss.

Averaging over 10 runs, the problem was solved to optimality in 2.338 seconds. The resulting solution placed each VOIP call on the direct link with (3,1) FEC as expected. This provides evidence that the approach is scalable to a reasonable number of flows. This is because we do not expect to plan over all flows in the system: we only expect specifications for flow requirements to be given for a subset of the flows.

Vignette 4: Example

This vignette has the network characteristics as described in Example 1, and the same flows.

Averaging over 10 runs, the problem was solved to optimality in 0.1888 seconds. The resulting solution placed the FTP flow on the direct link as expected, and rerouted the video flow with (3,1) FTP on the links 9 to 11, and then 11 to 3. This demonstrates rerouting and application of FEC as expected.

Conclusion

As hardware and controls for network science become more sophisticated, we are able to build on such successes to solve problems with more abstract specifications. Given mission level specifications such as deadline requirements or cumulative loss and delay requirements, naive approaches which maximize network usage will no longer be sufficient. In this paper, we outline a method for finding network configurations which conform to higher level specifications.

Our contributions are as follows. We have defined the network configuration planning problem, in which a planner must allocate bandwidth on the network and assign configurations for flows with deadline, bandwidth, delay and loss specifications. We have provided an encoding of the problem as a constraint program. This encoding was implemented and tested with an in-house solver, and we have reported on preliminary results, including qualitative behavior for a simple scenario, as well as computational time needed for small benchmarks.

There are several outstanding issue to be addressed in future work. While the current encoding is for a deterministic network configuration problem, we are implementing an extension which allows for probabilistic uncertainty in flow requirements and network conditions. The encoding will allow specifications for bounds on the probability of non-conformance with flow requirements, using risk allocation methods (Ono and Williams 2008; Fang, Yu, and Williams 2014). To improve the scalability of the solution algorithm, we will also leverage large neighborhood search (LNS) (Pisinger and Ropke 2010), a local search method which has been key to improvements in scalability for VRPs.

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency under Contract No. HR0011-15-C-0098.

References

Ben-Ameur, W., and Kerivin, H. 2005. Routing of uncertain traffic demands. *Optimization and Engineering* 6(3):283–313.

Bent, R., and Van Hentenryck, P. 2004. A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science* 38(4):515–530.

Fang, C.; Yu, P.; and Williams, B. C. 2014. Chance-constrained probabilistic simple temporal problems. In *AAAI*, 2264–2270.

Hall, R. 2016. Vehicle routing software survey. *OR/MS Today* 43(1).

Johnston, M.; Lee, H.-W.; and Modiano, E. 2013. Robust network design for stochastic traffic demands. *Journal of Lightwave Technology* 31(18):3104–3116.

Kilby, P., and Shaw, P. 2006. Vehicle routing. *Foundations of Artificial Intelligence* 2:801–836.

Muscettola, N.; Nayak, P. P.; Pell, B.; and Williams, B. C. 1998. Remote agent: To boldly go where no ai system has gone before. *Artificial Intelligence* 103(1-2):5–47.

Ono, M., and Williams, B. 2008. Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, 3427–3432. IEEE.

OpSat-v3. 2018. Optsat-v3. In *To be submitted to IJCAI*.

Pisinger, D., and Ropke, S. 2010. Large neighborhood search. In *Handbook of metaheuristics*. Springer. 399–419.

van Hoeve, W.-J., and Katriel, I. 2006. Global constraints. *Foundations of Artificial Intelligence* 2:169–208.