

Distributed Pseudo-tree Optimization Procedure Formulation In Clingo

Luming Wu

2017-12-20

1 Paper

Solving Distributed Constraint Optimization Problems Using Logic Programming is a paper wrote by Tiep Le, Tran Cao Son, Enrico Pontelli, William Yeoh from New Mexico State University. In the paper, they explained the method to use logic programming, Answer Set Programming(ASP), in implementing the Distributed Pseudo-tree Optimization Procedure(DPOP) for solving Distributed Constraint Optimization Problems(DCOP). And they provided a few benchmarks to demonstrate the benefit of using logic programming. However, the paper focused on the procedure in modeling ASP, some of my ideas also came from the paper *A Scalable Method for Multi-agent Constraint Optimization* by Adrian Petcu and Boi Faltings to help explain the algorithm.

2 Report

There are three goals in this report:

- (1) Explain how to formulate a distributed constraint optimization problem in Clingo's Answer Set Programming.
- (2) Demonstrate some of the potential problems in using Clingo.
- (3) Share some thoughts about the paper.

3 Distributed Pseudo-tree Optimization Procedure(DPOP)

3.1 Installiation

1. The latest clingo can be downloaded from:

<https://sourceforge.net/projects/potassco/files/clingo/>

2. After installiation, add the root directory of clingo.exe to the path for convenience.

3. In the root directory, there should be clingo.exe and clingo-python.exe. clingo-python.exe is required for simulating a distributed system. Therefore, Python 2.7 installiation is also required. Clingo requires a different version of Python 2.7, it is recommended to check it in the README of the root directory.

Here is a link for the Python used in this project:

<https://www.python.org/ftp/python/2.7.13/python-2.7.13.amd64.msi>

4. It is recommended to add Python 2.7 into the path. Although it is very likely that Python 2.7 will run into problem with Python 3. However, there was no conflict throughout the project. In the worst case, uninstall Python 3.

3.2 Command

The project will be using the following clingo command:

```
clingo-python [ASP_FILE] [PYTHON_CONTROL_FILE] -outf=3
```

For convenience, the project source code has a command commented in each .lp ASP file.

3.3 Answer Set Programming

This section will briefly explain the ASP that is used in the project.

DCOP Translation

```
variables(variable).
```

The set of variables, which includes current agent's variable and context variable. `variable` is a string constant passed by the controller, if there are `n` variables, the controller will ground this fact `n` times with different variable each time.

```
domains(variable, value).
```

The set of domains, which represents the values that a specific variable can be assigned. Both `variable` and `value` are string constants, this fact will be grounded multiple times to represent the whole domain.

```
1 {variable_assignment(Variable, Value) : domains(Variable, Value)} 1 :-  
    variables(Variable).  
assignment(Variable, Value) :- variable_assignment(Variable, Value).
```

The first line means for each variable `Variable`, given a list of possible values `Value` from the domain, only assign 1 value to `Variable`. The second line means a variable assignment is also an assignment, this represents inheritance.

```
utility(Utility) :- Utility = #sum {
1 : assignment(var1, val1), assignment(var2, val2);
...
}.
```

Utility will be the sum of values that are derived from the given assignment. 1 is an example utility and the right hand side of the condition describes the constraint.

Subprograms

```
#program program_name(argument, ...).
```

Because the project will have a Python controller running ASP, the controller will decide what has to be grounded. `#program` is used to divide the program into subprograms so that the controller only use ASP that is in the scope of the subprogram to ground. `program_name` is used to identify the program and arguments are the values passed from the controller. Example:

```
#program marriage(arg1, arg2).
husband(arg1).
wife(arg2).
```

External Grounding

```
#external fact(argument, ...).
```

External grounding is used when some facts are sometimes true and sometimes false. Example:

```
#external today(Date) : date(Date).
final_date(12/18/2017).
final_date(12/20/2017).
have_final :- today(Date), final_date(Date).
```

Notice that `Date` is actually a variable, and `today` has a condition `date(Date)`, which is grounded before. It is required because there are infinite possible dates for `today`, and `Date` could be anything like number and name. The program needs to know the domain of a fact that could be true or false.

Show

```
#show answer/arity
```

ASP will return a very large answer set when the domains and variables are very large. `#show` is used to filter answers that are not needed. For example, only the assignments are wanted:

```
#show assignment/2
```

3.4 Distributed System

This section will explain how this project can be extended to distributed programming.

Assumption

Although there are many different distributed database designs that can optimize this procedure. The idea is that the main node will communicate with all relevant servers as part of Distributed Depth-First Search(DFS). The servers will store data and/or receive data from the main node and/or other servers. The servers will then compute the UTILITY message and send it to the target server. When the main node as the root finalized the optimal assignment, it can inform other servers about the decision by DFS for VALUE propagation.

Server

Initially, each server will receive the job request and the relevant data in JSON. Example:

```
def get_domains():  
    return [("var1", ["rice", "noodle"]),  
            ("var2", ["chicken", "beef"]),  
            ("var3", ["java", "c++"]),  
            ("var4", ["server", "game"])]
```

The server will create a process locally to run `clingo-python.exe`, which also run a Python controller. The server will first create the empty UTILITY message table with all dimensions represented. The server will store the output temporarily just to pass it to the next process that finds all the possible assignments and utility given the context assignments. The process will only output the assignment that has the largest utility for each context assignments. When the server computed the UTILITY message, the message will be sent to the target server.

Data Type

The JSON data are made up of strings and integers. As a result, variables and values are represented by atoms and only the utilities are represented by number. Although variables can also be represented by numbers, this project will stick with atoms.

3.5 Python Controller

clingo-python API

clingo-python has a few API calls that could help grounding ASP programs. However, the API is relatively small and there are many limitations. The general limitation is anything that is grounded can't undo, unless `#external` is used. Although `#program` is a feature, it actually mean subprogram, they are additional program calls on top of the base program.

Ground Method

```
prg.ground([("program_name", [arguments,...])])
```

The ground method is the basic method for grounding a program. The method accepts a list of a tuple that has program name and argument list. However, it is tested that the method will only accept one tuple. Therefore, grounding multiple times is recommended. Program name refers to the program name for `#program`, the default base program is named 'base '. In the argument list, both string and number are accepted. But the number of arguments must match the arity in `#program` definition. The arguments will be represented by macro constants in the program, their names also depend on `#program` definition.

External Methods

```
for variable, value in assignment:
    prg.assign_external(Function("assignment", [String(variable), String(value)]), True)
optimal_symbols = []
with prg.solve(yield_=True) as handle:
    for model in handle:
        optimal_symbols.append(model.symbols(shown=True))
for variable, value in assignment:
    prg.assign_external(Function("assignment", [String(variable), String(value)]), False)
```

`assign_external` temporarily assign truth value to a `#external` statement. `Function()` will construct a Symbol which could have a name and an argument list. In the argument list, there can be `String(...)`, `Number(...)`, or more Symbols. A Symbol can be viewed as a fact that has a name and a list of arguments in the parenthesis. In addition, there is another method named `release_external(...)`. It takes similar arguments, but it is to make the program never consider the Symbol is true or false.

Add Method

```
prg.add("program_name", ["argument_name"], "non_ground_program")
```

The `add(...)` method is a powerful method that allows programmer to add a subprogram at runtime. It is similar to creating a subprogram in the ASP file. Compare to the other two methods, `add(...)` can actually parse the non-ground program. That means `add(...)` can add integrity constraints.

3.6 Project Files

Git Repository

<https://github.com/LumingWu/DPOP-implementation>

File Description

`stage3_utility_message.lp`: An ASP file that computes the dimensionality of the UTILITY message.

`stage3_utility_message_controller.lp`: A Python controller for the file `stage3_utility_message.lp`. It passes the JSON data into the ASP and it receives the UTILITY message table.

`stage3_optimization.lp`: An ASP file that tries to find the optimal assignment and utility for every UTILITY message combination.

`stage3_optimization_controller.py`: A Python controller for the file `stage3_optimization.lp`. It passes the JSON data and the UTILITY message table into the ASP and it will select the optimal answer set to output.

4 Possible Clingo Problems

Clingo is a powerful ASP tool, but it is not flawless.

Python Version

clingo-python is still using a specific version of Python 2.7. It creates conflict with Python 3, it becomes a problem when servers are running in Python 3. Python 2 is getting less official support, the problem can only get worse.

Optimization

Although Clingo has multiple ways to ground a program, it is far from enough for programmers to customize and decide a more optimal communication between controller and ASP. In addition, the communication between controller and ASP is in a string or a structure that is not as solid as JSON. It was expensive to have the controller to parse an ASP output or parse the input into an ASP structure.

Filter

Clingo does not support program specific answer display. `#show` statements are shared among all programs, there is no customization for answer set output.

5 Final Thought

Although this report is about how to translate DPOP to Clingo, it seems Clingo is not an ideal tool for the task. In reality, decimal numbers are more likely to be considered as utility of a constraint. And in the field of machine learning, probability also require the support of decimal number. About the paper, the paper gave a very general idea on the implementation. Through working on the project, it seems there are problems like constraint formulation, distributed database design, and data structure. Nevertheless, the use of dynamic programming in solving problems like DCOP was definitely a good direction.