September 25, 2018
Recruit Communications Co., Ltd.

Releasing Domain Specific Language for QUBO Auto-building in Solving "Combination Optimization Problems" with Annealing Machine as OSS on Thursday, September 25

Recruit Communications Co., Ltd. (Headquarters:Chuo-ku,Tokyo; Representative Director, President: Atsushi Shimizu; hereinafter referred to as "Recruit Communications") developed PyQUBO, a domain-specific language to automatically build QUBO* in attempt to promote commercial use of annealing machine. It releases PyQUBO as OSS from GitHub on Thursday, September 25,2018.

* QUBO (quadratic unconstrained binary optimization) is a mathematical formula required for "combination optimization problems " in using an annealing machine.

About PyQUBO Development

●Background

There are many "combination optimization problems" in the business world that require much more than simple computation, and thus it has been impractical to use conventional computers in solving these problems. Recruit Communications began research on quantum annealing[*1] technology in the attempt to use it in marketing communications, and has been developing theories of data analytics using annealing technology[*2] such as quantum annealing and use cases with production data and discussing the implementation of our software program.

●PyQUBO

Leveraging insight and learning from discussions of annealing technology and its use cases, Recruit Communications successfully developed PyQUBO, a domain specific language (DSL) that is capable of auto-generation of QUBO, which is necessary for solving "combination optimization problems" using an annealing machine.

In order to solve "combination optimization problems" with an annealing machine the user must describe the problem as a mathematical formula and convert it to useable code. This conversion process is complex and often leads to software defects.

With PyQUBO, all what the user needs to do is to write an objective function in the same format as the mathematical formula of their combination optimization problem; The QUBO compiler processes it to run on an annealing machine, becoming simple and intuitive.

< Without PyQUBO >

The statement using the result of manual calculation results in lower readability.

```
def index(i, j):
    return i*n_city + j

A = 4.0
Q = defaultdict(float)
for i in range(n_city):
    for j in range(n_city):
        Q[index(i, j), index(i, j)] -= 2*A

for i in range(n_city):
    for j in range(n_city):
        for k in range(n_city):
            if j != k:
                Q[index(i, j), index(i, k)] += 2.0 * A
                Q[index(j, i), index(k, i)] += 2.0 * A

for u in range(n_city):
    for v in range(n_city):
        for i in range(j, n_city):
            Q[index(u, i), index(v, (i+1)%n_city)] += d(u, v)
```

< With PyQUBO >

The statement using the mathematical formula as is automatically generates QUBO.

```
# Write TSP in DSL
x = Matrix('x', n_city, n_city)
distance = Sum(0, n_city, lambda u: Sum(0, n_city, lambda v:
             Sum(0, n_city, lambda j: d(u, v) * x[j, u] * x[(j+1)%n_city, v])))
const_1 = Sum(0, n_city, lambda v: (Sum(0, n_city, lambda j: x[j, v]) - 1)**2)
const_2 = Sum(0, n_city, lambda j: (Sum(0, n_city, lambda v: x[j, v]) - 1)**2)

# Construct Hamiltonian and compile it
A = Param("A")
H = distance + A * (const_1 + const_2)
model = H.compile()

# Generate QUBO
qubo, offset = model.to_qubo(params={'A': 4.0})
```

$$H = \underbrace{\sum_u^n \sum_v^n d_{uv} \sum_j^n x_{j,u} x_{j+1,v}}_{\text{Distance of the route}} + A \underbrace{\sum_v^n \left(\sum_j^n x_{j,v} - 1 \right)^2}_{\text{Constraint 1}} + A \underbrace{\sum_j^n \left(\sum_v^n x_{j,v} - 1 \right)^2}_{\text{Constraint 2}}$$

PyQUBO example of traveling salesman problem

Recruit Communications Co., Ltd.

PyQUBO Open Source Version

Recruit Communications releases “PyQUBO” as open source software on September 25, available from Github in order to drive commercial use of an annealing machine. Annealing machines have been attracting a lot of attention recent years.
Recruit Communications is partnering with academia, research institutions, as well as developers for collaboration and joint development of annealing machine use cases in cutting-edge areas. By making PyQUBO available as OSS Recruit Communications is going to help accelerate the creation of annealing machine use cases. “PyQUBO” uses Python, which is a programming language widely adopted in studies of statistical analyses and annealing machines. Because of its high penetration the user of PyQUBO will be able use annealing machine without specific experience or expertise. Further enhancements of PyQUBO can be expected via contributions of the open source engineering communities.

PyQUBO GitHub Site: https://github.com/recruit-communications/pyqubo

Announcement at Qubits North America 2018
The detail of PyQUBO development will be presented at “D-Wave Users Conference Qubits North America 2018” at Knoxville, Tennessee on Thursday, September 25, 2018.

Research on Commercial Deployment of Annealing Machine

Recruit Communications develops a wide range of marketing solutions for customer acquisition, web marketing, media production and advertisement for the Recruit Group companies. In digital marketing personalization is essential and there are many matching problems of “the service user x content x advertisement.” Type. Conventional computers are inadequate in solving such problems in a reasonable amount of time. Recruit Communications has been studying annealing machine-based solutions to substantially reduce the time required for solving “combination optimization problems” in a commercial environment. Recruit Communications’ mission is to develop advanced marketing technologies with which optimal, high-quality services and advertisement derived from the combination of personal preferences and multiple attributes in delivering outstanding user experiences to the Recruit Group companies as well as their customers.

Glossary

*1 Quantum Annealing
Quantum annealing is a technique to solve optimization problems by simulating physical dynamics in which quantum effects are reflected. It is said to be advantageous in solving non-linear optimization problems that have many local optima and effective in machine learning in which natural language processing and clustering processing in marketing technology are often mathematically formulated as optimization problems.

*2 Annealing Technology
It is a generic algorithm to obtain a solution of the optimization problem by “annealing” on a computer where a state with an organized structure maintaining a minimal energy is created.

*3 DSL (Domain-Specific Language)
DSL or domain-specific language. Unlike generic programming languages such as Java and Python, DSL refers to a specialized programming language for solving problems of a specific domain.

Contact details for inquiries regarding this press release:
https://www.rco.recruit.co.jp/contact/