ISSN 2320 - 2602
Volume 4 No.8, August 2015
International Journal of Advances in Computer Science and Technology
Available Online at http://www.warse.org/IJACST/static/pdf/file/ijacst01482015.pdf

Using Parallel and Distributed Computing Paradigm for Optimization of Ultimate Pit Limit Determination

Noshab Gul, Hammad Ali Zai
Department of Computer Systems Engineering, University of Engineering and Technology Peshawar, Pakistan
cse.noshab@gmail.com

ABSTRACT

Efficient parallel computing techniques can make the solution of computationally challenging optimization problems traceable. Optimization problems from varied disciplines can be solved more efficiently through parallel and distributed computing. Optimization of Ultimate Pit Limit (UPL) determination is an important problem of mining engineering. We used a parallel and distributed computing architecture based on Python Remote Objects and Python Optimization Modelling Objects (PyRO-PyOMO), for UPL determination problem. The results show that exploiting parallelism help in achieving 70 % speedup in computation time on various mining datasets. We find that the programming effort associated with efficient parallelization of optimal ultimate pit limit determination using PyRO-PyOMO architecture is highly non-trivial. A similar parallel computing model can be used for the various mathematical models and optimization methods used to solve other optimization problems as well.

Key words : Parallel computing, optimization, mining, Python Remote Objects (PyRO).

1. INTRODUCTION

Mining is about extraction of minerals from the surface of earth, economically and safely. Open pit mining is one of the surface mining methods, where a big excavation is made to extract the minerals [1]. Open pit mine is an inverted stepped cone like structure. Normally, the valuable minerals known as ore are surrounded by a non-valuable host rock known as waste. Some portion of the waste is also removed in order to access the ore and make the open pit stable.

Determination of the optimum UPL of the mine is a fundamental problem in open pit design. To maximize the difference between total extraction cost of the ore/waste and the extracted ore, the determination of the optimal pit contour is important. For evaluation of the economic potential of the mineral deposit the necessary information is provided by the determination of optimal pit contour [1].

An open pit is designed in such a way that the profits are maximized, resulting excavation is stable and health, safety and environmental standards are ensured [2]. This will result in an efficient design for extraction of minerals in the form of defining optimal UPL. Optimal UPL determination problem can be solved using mathematical modelling and optimization techniques. Mathematical modelling formulation of UPL determination is represented as follows, using the notions where:

V = Set of all blocks that can be mined.
A = Set of pairs (*i*, *j*) of blocks such that block *j* is a neighboring block to *i* that must be removed before block *i* can be mined.
c_i = Cost of mining and processing block *i*.
r_i = Revenue obtained from block *i*.
p_i = Profit obtained by mining and processing block *i* (i.e., $p_i = r_i - c_i$).

$$x_i = \begin{cases} 1, & if block\ i\ is\ mined \\ 0, & otherwise \end{cases}$$

Mathematical optimization model for UPL determination, i.e. maximizing the total profit, is therefore given by [3]:

$$\max \sum_{i \in V}^{n} p_i x_i$$

subject to

$$x_i \leq x_i \qquad (i, j) \in A$$
$$x_i \leq \{0, 1\} \qquad i \in V$$

Heavy mathematical calculations and large data sets are involved in optimal UPL determination problems. This results in long computational time using a single computer. Therefore, these problems are computationally challenging and efficient parallel and distributed computing techniques can make its solution traceable. Parallel or distributed computing is feasible to solve this problem by reducing the time needed to obtain the solution. Because, computational performance can be boosted substantially by splitting the workload between cores and (or) processors.

Various open pit optimization algorithms have been reviewed. Many researchers have solved this problem by applying their own optimization techniques. Existing optimization are also applied to solve UPL problem. However instead of all the previous research, scholars are still in search of superior models, algorithms, and efficient computational strategies.

2. PARALLEL COMPUTING PARADIGM FOR ULTIMATE PIT LIMIT DETERMINATION

A parallel computing based software architectures is proposed for optimal ultimate pit limit determination. The proposed architecture is based on PyRO-PyOMO. Flowchart showing the methodology adopted in undertaking this research endeavour, presented in Figure 1.

General mathematical models for optimal UPL determination are developed in an open source Python based software package i.e. Common Optimization Python Repository (COOPR). PyOMO is a part of COOPR package. PyOMO's modelling objects are embedded within python with a rich set of supporting libraries. PyOMO is used to define abstract models for optimal UPL determination, create concrete instances of UPL mathematical models and solve those instances with standard solvers.

COOPR provides an interface that supports parallel solver execution and distributed optimization. Multiple UPL models are solved concurrently in parallel on a single machine, over a cluster and over NEOS (Network enabled optimization server). NEOS is used as it provides an interface to solve UPL models on remote resources.

2.1 PyRO- PyOMO Based Software

COOPR supports distributed computing via one of its components and Python package named PyRO. COOPR includes a flexible framework for applying optimizers to analyze PyOMO models [4].

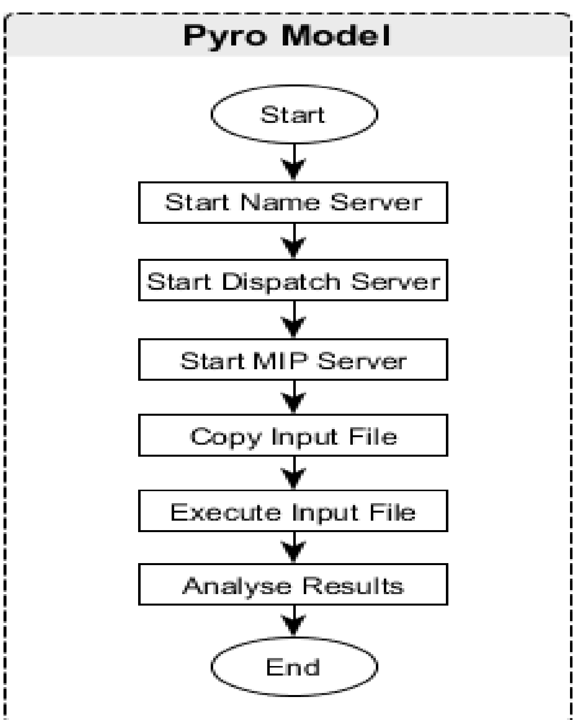

Figure 1: Flowsheet of PyRO-PyOMO architecture

Cooper includes two components that manage the execution of optimization solvers. Solver performs optimization while solver manager i.e. PyRO, supports parallel execution of solvers. This architecture support parallel and distributed execution of multiple UPL models

Through solver parallelization this algorithm significantly reduces solution time for problem containing millions of blocks. This software model works through two ways:

1. Creating and using a client and multiple solvers in parallel on a single, multi-core compute server
2. Distributed solves under PyRO, over clusters or remote NEOS resources

3. RESULTS AND DISSCUSSION

MineLib datasets are used to generate mathematical models of multiple scenarios and instances. MineLib 2011 includes a library of publicly available test problem instances for UPL problem. The data comes from real-world mining projects and simulated data [5]. Abstract and Concrete modelling techniques are used to model and solve optimal UPL determination problem in python. These abstract and concrete models are defined through PyOMO. These problem instances are solved through multiple open source solvers like GLPK and CBC etc. Initially the problem was solved using sequential solution strategy, on a local multi-core machine. Comparison of computational time of using CBC solver on local and remote machine i.e. NEOS server is presented in Table 1, 2 and 3. Similarly graphs of these results are presented in Figures 2 - 5.

Table 1: Comparison of Computational Time and Process Time Ratio of CBC Solver Running Locally and Remotely On NEOS Server

S. #	Dataset Details	Computational Time (Seconds)		
	Name of Dataset	**Sequential / Local**	**Remote / Distributed**	**Process Time Ratio**
1	newman1	1.38	1.87	0.74
2	sm2	70.16	1.79	39
3	zuck_small	119.33	3.00	40
4	Kd	234.37	2.33	101
5	Marvin	214.87	2.89	74
6	p4hd	5,621	25.50	220
7	w23	6,945	86.38	80
8	zuck_large	10,290	77.57	133
9	zuck_medium	12,766	104.47	122
	Average	**4,029**	**34**	**90**

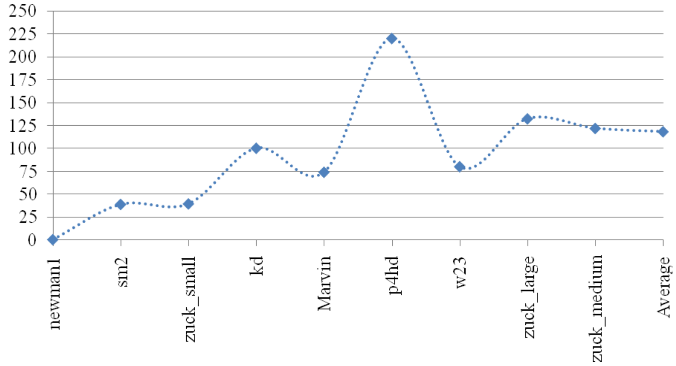

Figure 2: Graph of Process Time Ratio between CBC Sequential and CBC NEOS Server

Table 2: Comparison of Computational Time and Process Time Ratios of Using CBC Solver and Sequential and Parallel Models, Solving Multiple Instances of UPL Problem

S. #	Dataset Details	Computational Time of 3 Instances (Seconds)		
		Sequential / Local	**Remote / Distributed**	**Process Time Ratio**
1	newman1	3.49	1.69	2.06
2	sm2	284	123	2.31
3	zuck_small	302	162	1.86
4	Kd	628	323	1.95
5	Marvin	844	346	2.44
6	p4hd	6,658	3,512	1.90
7	w23	16,216	9,227	1.76
8	zuck_large	21,108	9,218	2.29
9	zuck_medium	27,432	13,567	2.02
	Average	**8,164**	**4,053**	**2.07**

Table 3: Log of Computational Time of Using CBC Solver and Sequential and Parallel Model, Solving Multiple Instances of UPL Problem

S. #	Name	Log (PyRO – Multiple Instances)
1	newman1	2.06
2	sm2	2.31
3	zuck_small	1.86
4	Kd	1.95
5	Marvin	2.44
6	p4hd	1.90
7	w23	1.76
8	zuck_large	2.29
9	zuck_medium	2.02
	Average	**2.07**

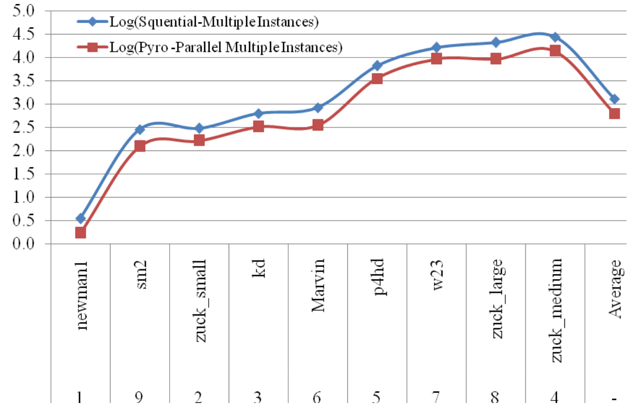

Figure 3: Graph of Log of Computational Time of Running Multiple Instances

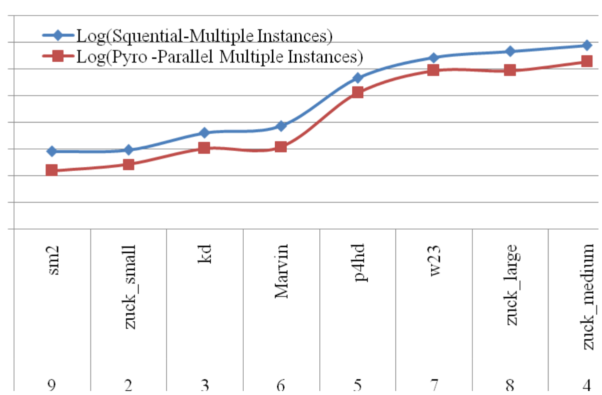

Figure 4: Graph of Log of Computational Time of 8 MineLib Datasets Excluding Newman1 and Running Multiple Instances

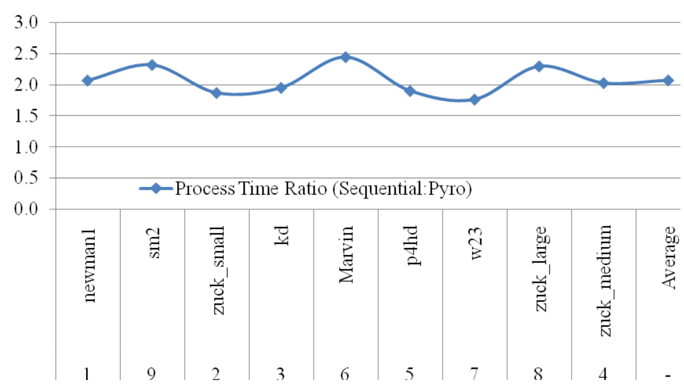

Figure 5: Graph of Process Time Ratio of Sequential and Parallel Software Models

These results indicate very clearly that distributed computing is helpful in solving such problems on remote powerful machines, more efficiently. Using 9 MineLib data set for solving optimal determination of UPL problem on local machine using cbc solver, is on average 90 times more time consuming as compared to solve them on remote machine i.e. NEOS in our case using cbc solver. Variation in computational time fluctuates considerably between 1.38 to 12,677.

These results indicate that Pyro-Pyomo based parallel computing model for solving optimal UPL determination problem are on average 2.07 times computationally more efficient as compared to the sequential solution of multiple instance.

4. CONCLUSIONS AND RECOMMENDATIONS

Computationally challenging ultimate pit limit determination problem can be made traceable through parallel computing and programming. Efficient parallel computing and programming techniques enables the efficient use of hardware resources of multi-core processors and high performance graphical processing units (GPUs). Therefore, a parallel computing software model was developed for mathematical modeling and optimization of UPL problem.

The proposed system is based on fundamental concepts of parallel and distributed computing, that enable the execution of solver processes on distinct cores of single work station, across multiple workstations and provide a flexible mechanism for communication among them. Major contributions of this research endeavor are:

- Python based parallel and distributed programming system have shown an improvement in computational time for solving ultimate pit limit determination problem relating open pit mine planning and design optimization.
- Compared to other parallel and distributed programming systems, the proposed systems are integrated with the dynamically-typed scripting python language and are relatively simple in implementation and usage.
- It has the ability to be used as a prototyping tool for new mathematical modeling and optimization models.
- Execution models of parallel programming if managed properly can give much better results, as for as time complexity reduction is concerned in mine planning and design optimization problems.

The strands of work initiated and presented, can be extended and a future investigation is proposed in the following domains:

- This work is based on the implementation of software architecture which provides high level parallelism. However, low level parallelism (task level parallelism) can be incorporated using other efficient open source mathematical modeling and optimization tool kits like SCIP.

REFERENCES

1. Laurich, R.. **Planning and design of surface mines**. B. kennedy ed. *Surface Mining, Chapter 5.2*. Port City Press, Baltimore. Pp 465-469. 1990.
2. Peter Kwagyan Achireko. **Application of Modified conditional simulation and artificial neural networks to open pit optimization.** Chapter 1 Thesis, 1-2. 1998.
3. Henry Amankwah. **Mathematical Optimization Models for Open-Pit Mining.** PhD Thesis. Linköping University Sweden. 2011.
4. Daniel Espinoza. Marcos Goycoolea. Eduardo Moreno. Alexandra Newman. **MineLib: a library of open pit mining problems.** Ann Oper Res. 206:93–114 DOI 10.1007/s10479-012-1258-3. 2013.
5. Alexandra M. Newman, Enrique Rubio, Rodrigo Caro, Andres Weintraub, Kelly Eurek. **A review of operations research in mine planning.** DOI 10.1287/inte 1090.0492 *INFORMS.* 2010.