Easy Modeling of Open Pit Mining Problems via Constraint Programming

Broderick Crawford[1,3], Ricardo Soto[1,2], Carolina Zec[1], Eric Monfroy[4], and Fernando Paredes[5]

[1] Pontificia Universidad Católica de Valparaíso, Chile
[2] Universidad Autónoma de Chile, Chile
[3] Universidad Finis Terrae, Chile
[4] CNRS, LINA, Université de Nantes, France
[5] Escuela de Ingeniería Industrial, Universidad Diego Portales, Chile
{ricardo.soto,broderick.crawford}@ucv.cl,
carolina.zec.s@mail.pucv.cl,
eric.monfroy@univ-nantes.fr,
fernando.paredes@udp.cl

Abstract. The open pit mining problem aims at correctly identifying the set of blocks of a given mine to be extracted in order to maximize the net present value of the production. During the last years, different techniques have been proposed to solve mining problems, which range from the classic mathematical programming to more recent ones such as the metaheuristics. In this paper we illustrate how this problem can easily be solved by a relatively modern and declarative programming paradigm called constraint programming.

Keywords: Constraint Programming, Metaheuristics, Manufacturing Cell Design.

1 Introduction

Open pit mining is a mineral extraction technique in which the orebody is reached by opening a large ground surface along a mine. The orebody is commonly discretized and modeled as a set of blocks. The goal of the classic open pit mining problem is to correctly identify the blocks of the mine to be extracted in order to maximize the net present value of the production. There exist varied versions of the problem, that mainly consider different kinds of constraints related to the distribution of blocks along the mine, to the production planning, and to the processing plant and mining capacity. During the last thirty years, different techniques have been proposed to solve mining problems. Some examples are the classic linear and mathematical programming [4,10,1], and metaheuristics such as the genetic algorithms [19,8].

In this paper, we focus on solving open pit mining problems (OPMP) via Constraint Programming (CP). CP is a programming paradigm for the efficient solving of constraint-based and optimization problems that has successfully been

C. Stephanidis (Ed.): HCII 2014 Posters, Part I, CCIS 434, pp. 519–522, 2014.

employed in several application domains, different examples can be found in rostering [15], manufacturing [18], games [12,13] and in engineering design [11]. In CP, a problem is modeled by representing the unknowns of the problem as variables and the relations among them as constraints. This mathematical model is then encoded in the language of a CP search engine, which is responsible for search a result. In this paper we illustrate how this problem can easily be modeled by using CP, in particular via the MiniZinc modeling language [9]

From a user perspective, solving problems via CP has an important advantage with respect to metaheuristics: there is no need to design a specific algorithm to solve the problem, the user only models the problem and the solver provides a result. In this paper, we illustrate how this problem can easily be solved by using CP, in particular via the MiniZinc modeling language [9].

2 Modeling the OPMP in CP

CP provides an important feature to the user: there is no need to design a specific algorithm to solve a given problem, the user only models the problem and the search engine search for a result. This can clearly be contrasted to the use of metaheuristics, where the user must implement practically from scratch a new algorithm following some pre-established patterns. In CP, it suffices to encode the mathematical model in the language of the solving engine. In the following we illustrate how the OPMP can easily be solved by using the CP paradigm. For space reasons we consider only the objective function and a single constraint, the complete model can be seen in [2].

The objective function aims at maximizing the total profit from the mining process in a given period of time, where t is the time period index and T is the number of time periods considered; n is the block index and N is the total number of blocks. C_n^t is the net present value obtained from mining block n in time t; and x_n^t is a binary decision variable which is set to 1 if the block is mined and 0 otherwise.

$$\texttt{maximize}\ \ Z = \sum_{t=1}^{T} \sum_{n=1}^{N} C_n^t x_n^t \tag{1}$$

The corresponding MiniZinc code is shown below, which is a straightforward mapping from the mathematical definition of the objective function. A `z` value captures the double summation which is then launched via the `solve maximize` instruction.

```
1. var int: z = sum(n in 1..N,t in 1..T)
2.                  (C[n,t] * x[n,t]);
3. solve maximize z;
```

The total tons of material to be exploited are restricted by processing and mining capacities. The total material mined, involving ore and waste must respect the given upper bound as stated in Eq. 2; where To_n is amount of ore in block n, Tw_n is amount of waste in block, and MC^t_{max} is the maximum material, including waste and ore, to be mined in period t.

$$\sum_{i=1}^{n} (To_n + Tw_n) x^t_n \leq MC^t_{max} \qquad t \in (1, 2, ..., T) \tag{2}$$

```
constraint
   forall(t in 1..T) (
     sum(n in 1..N)
       ((To[n] + Tw[n]) * x[n,t]) <= mcmax[t]
   );
```

3 Conclusion

In this paper, we have illustrated how the OPMP can be modeled by using the CP paradigm. In CP, it suffices to encode the corresponding mathematical model in the solver language in contrast to metaheuristics that require to implement a specific algorithm to solve the problem.

A continual research direction in CP is about facilitating the user modeling and experimentation phases, such as for instance to propose easy-to-use modeling languages [17,3] and modeling features [16]. The use of autonomous search [14,6,7,5] for solving this problem will be an interesting direction to pursue as well.

Acknowledgements. Ricardo Soto is supported by Grant CONICYT/ FONDECYT/INICIACION/11130459, Broderick Crawford is supported by Grant CONICYT/FONDECYT/REGULAR/1140897, and Fernando Paredes is supported by Grant CONICYT/FONDECYT/REGULAR/1130455.

References

1. Boland, N., Dumitrescu, I., Froyland, G., Gleixner, A.: LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity. Computers & Operations Research 36(4), 1064–1089 (2009)
2. Caccetta, L., Hill, S.: An application of branch and cut to open pit mine scheduling. Journal of Global Optimization 27(2-3), 349–365 (2003)
3. Chenouard, R., Granvilliers, L., Soto, R.: Model-Driven Constraint Programming. In: Proceedings of the 10th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming (PPDP), pp. 236–246 (2008)
4. Chicoisne, R., Espinoza, D., Goycoolea, M., Moreno, E., Rubio, E.: A New Algorithm for the Open-Pit Mine Production Scheduling Problem. Operations Research 60(3), 517–528 (2012)

5. Crawford, B., Castro, C., Monfroy, E., Soto, R., Palma, W., Paredes, F.: Dynamic Selection of Enumeration Strategies for Solving Constraint Satisfaction Problems. Romanian Journal of Information Science and Technology 15(2), 106–128 (2012)
6. Crawford, B., Soto, R., Castro, C., Monfroy, E.: Extensible CP-based autonomous search. In: Stephanidis, C. (ed.) Posters, Part I, HCII 2011. CCIS, vol. 173, pp. 561–565. Springer, Heidelberg (2011)
7. Crawford, B., Soto, R., Montecinos, M., Castro, C., Monfroy, E.: A framework for autonomous search in the ecl^{i}ps^e solver. In: Mehrotra, K.G., Mohan, C.K., Oh, J.C., Varshney, P.K., Ali, M. (eds.) IEA/AIE 2011, Part I. LNCS, vol. 6703, pp. 79–84. Springer, Heidelberg (2011)
8. Denby, B., Schofield, D.: Open-pit design and scheduling by use of genetic algorithms. Transactions of the Institution of Mining and Metallurgy, Section A: Mining Industry 26, A21–A26 (1994)
9. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.R.: MiniZinc: Towards a Standard CP Modelling Language. In: Bessière, C. (ed.) CP 2007. LNCS, vol. 4741, pp. 529–543. Springer, Heidelberg (2007)
10. Ramazan, S.: The new Fundamental Tree Algorithm for production scheduling of open pit mines. European Journal of Operational Research 177(2), 1153–1166 (2007)
11. Soto, R.: Controlling search in constrained-object models. In: Kuri-Morales, A., Simari, G.R. (eds.) IBERAMIA 2010. LNCS, vol. 6433, pp. 582–591. Springer, Heidelberg (2010)
12. Soto, R., Crawford, B., Galleguillos, C., Monfroy, E., Paredes, F.: A hybrid ac3-tabu search algorithm for solving sudoku puzzles. Expert Syst. Appl. 40(15), 5817–5821 (2013)
13. Soto, R., Crawford, B., Galleguillos, C., Monfroy, E., Paredes, F.: A Pre-filtered Cuckoo Search Algorithm with Geometric Operators for Solving Sudoku Problems. The Scientific World Journal, Article ID 465359 (2014)
14. Soto, R., Crawford, B., Monfroy, E., Bustos, V.: Using autonomous search for generating good enumeration strategy blends in constraint programming. In: Murgante, B., Gervasi, O., Misra, S., Nedjah, N., Rocha, A.M.A.C., Taniar, D., Apduhan, B.O. (eds.) ICCSA 2012, Part III. LNCS, vol. 7335, pp. 607–617. Springer, Heidelberg (2012)
15. Soto, R., Crawford, B., Monfroy, E., Palma, W., Paredes, F.: Nurse and Paramedic Rostering with Constraint Programming: A Case Study. Romanian Journal of Information Science and Technology 16(1), 52–64 (2013)
16. Soto, R., Crawford, B., Monfroy, E., Paredes, F.: Syntax Extensions for a Constrained-Object Language via Dynamic Parser Cooperation. Studies in Informatics and Control 21(1), 41–48 (2012)
17. Soto, R., Granvilliers, L.: The Design of COMMA: An Extensible Framework for Mapping Constrained Objects to Native Solver Models. In: Proceedings of IEEE ICTAI, pp. 243–250 (2007)
18. Soto, R., Kjellerstrand, H., Duran, O., Crawford, B., Monfroy, E., Paredes, F.: Cell formation in group technology using constraint programming and boolean satisfiability. Expert Syst. Appl. 39(13), 11423–11427 (2012)
19. Zhang, M.: Combining genetic algorithms and topological sort to optimize open-pit mine plans. In: Proceedings of the 15th Mine Planning and Equipment Selection, pp. 1234–1239 (2006)