Journal of Industrial Engineering and Management

doi:10.3926/jiem.2008.v1n2.p4-15 ©© JIEM, 2008 – 01(02): 4-15 - ISSN: 2013-0953

Variations in the efficiency of a mathematical programming solver according to the order of the constraints in the model

Rafael Pastor

Universitat Politècnica de Catalunya (SPAIN)

rafael.pastor@upc.edu

Received September 2008
Accepted November 2008

Abstract: It is well-known that the efficiency of mixed integer linear mathematical programming depends on the model (formulation) used. With the same mathematical programming solver, a given problem can be solved in a brief calculation time using one model but requires a long calculation time using another. In this paper a new, unexpected feature to be taken into account is presented: the order of the constraints in the model can change the calculation time of the solver considerably. For a test problem, the Response Time Variability Problem (RTVP), it is shown that the ILOG CPLEX 9.0 optimizer returns a ratio of 17.47 between the maximum and the minimum calculations time necessary to solve optimally 20 instances of the RTVP, according to the order of the constraints in the model. It is shown that the efficiency of the mixed integer linear mathematical programming depends not only on the model (formulation) used, but also on how the information is introduced into the solver.

Keywords: mixed integer linear mathematical programming, response time variability problem, combinatorial optimization

1 Introduction

Integer linear programming is a classical tool in practical operations research that can be applied to many problems (e.g. Salkin and Mathur, 1989) very effectively (e.g. in Corominas et al. 2008 it was applied to solve a real problem of a

doi:10.3926/jiem.2008.v1n2.p4-15 ©© JIEM, 2008 – 01(02): 4-15 - ISSN: 2013-0953

motorcycle assembly line and in Pastor et al. 2008 it was applied to solve the case of a woodturning company). The technique is well-known and reliable but it must be handled carefully. It is known that its efficiency depends on the model (formulation) used: with the same mathematical programming solver, a given problem can be solved in a brief calculation time using one model but requires a long calculation time using another. Therefore, as stated by Billionnet (1999, p. 105): "*Given a problem with a few dozen of variables one cannot be confident that integer programming will work until it has been tried on realistic instances*".

Several techniques have been used to improve the efficiency of this tool. A standard technique is the elimination of symmetries: Margot (2007), for example, presents techniques for handling symmetries in integer linear programs in which variables can take integer values, which extends previous research that dealt exclusively with binary variables. Tightening the definition of the data and introducing redundant constraints have also provided good results: Corominas et al. (2006), for example, demonstrated the importance of modelling, as well as the huge impact that redundant constraints and the elimination of symmetries have on the effectiveness of MILPs for solving the Response Time Variability Problem (RTVP), an NP-hard scheduling problem (Corominas et al., 2007); the total computation time taken to solve 20 instances dropped from 38,603 to 398 seconds and its practical limit to obtaining optimal solutions was increased from 25 to around 40 units to be scheduled.

This paper argues that the order of the constraints in a model can have a considerable effect on the time that a mathematical programming solver takes to solve a problem optimally. Lets us, for example, take three sets of constraints (A, B and C) of a problem to be solved. To introduce the sets of constraints in the mathematical programming solver in the order A-B-C, A-C-B, B-A-C, B-C-A, C-A-B and C-B-A is not indifferent and can cause its efficiency to vary considerably. This new, unexpected feature that must be taken into account in mathematical programming has not been presented previously (to the best of the authors' knowledge).

For an integer programming formulation of a test problem, RTVP, it is shown that the ILOG CPLEX 9.0 optimizer returns a ratio of 17.47 between the maximum and minimum calculation times needed to solve optimally 20 instances of the RTVP, according to the order of the constraints in the model: with one permutation of the

doi:10.3926/jiem.2008.v1n2.p4-15 ©© JIEM, 2008 – 01(02): 4-15 - ISSN: 2013-0953

sets of constraints the solver takes only 335 seconds, whereas with another one it takes 5,851 seconds. It is shown that the efficiency of the mixed integer linear mathematical programming depends not only on the model (formulation) used, but also on how the information is introduced into the solver.

The rest of the paper is organized as follows. Section 2 presents the RTVP. Section 3 describes the computational experiment carried out. Finally, Section 4 is devoted to the conclusions.

2 The test problem: the Response Time Variability Problem

The Response Time Variability Problem (RTVP) occurs whenever products, clients or jobs need to be sequenced so as to minimize variability in the time between the instants at which they receive the necessary resources. It was recently defined in the literature and first presented by Corominas et al. (2007), who proved that it is NP-hard.

This problem has a broad range of real-world applications. For example, it can be used in the automobile industry to sequence the models to be produced on a mixed-model assembly line (Monden, 1983). Other contexts in which the RTVP appears are the computer multi-threaded systems and network servers (Waldspurger and Weihl, 1995), the periodic machine maintenance problem (Bar-Noy et al., 2002), the scheduling of advertising slots for television (Brusco, 2008), the design of sales catalogs (problem introduced in Bollapragada et al., 2004) and the scheduling of waste collection (Herrmann, 2007).

The abovementioned applications are examples of a very common situation in which a resource must be used successively by different units and it is important to schedule them in such a way that the different types of units share the resource in some fair manner. The RTVP proposes a measure of fairness: to minimize the variability of the distance (measured, for example, in number of slot times) between any two consecutive units of the same product (event, job or client); i.e., to have the distances between any two given consecutive units of the same product as constant as possible.

The RTVP is formulated as follows. Let n be the number of products, d_i the number of units of product i $(i = 1,...,n)$ and D the total number of units

doi:10.3926/jiem.2008.v1n2.p4-15 ©© JIEM, 2008 – 01(02): 4-15 - ISSN: 2013-0953

$\left(D=\sum_{i=1}^{n} d_i\right)$. Let s be a solution of an instance in the RTVP that consists of a circular sequence of units $\left(s=s_1 s_2 \ldots s_D\right)$, where s_j is the unit sequenced in position j of sequence s. For all products i in which $d_i \geq 2$, let t_k^i be the distance between the positions at which the units $k+1$ and k of product i are found (i.e. the number of positions between the units, where the distance between two consecutive positions is considered equal to 1). As the sequence is circular, position 1 comes immediately after position D; therefore, $t_{d_i}^i$ is the distance between the first unit of product i in a cycle and the last unit of the same product in the preceding cycle. Let $\bar{t}_i$ be the average distance between two consecutive units of product i $\left(\bar{t}_i = D/d_i\right)$. For all products i in which $d_i = 1$, t_1^i is equal to $\bar{t}_i$. The objective is to minimize the $RTV = \sum_{i=1}^{n}\sum_{k=1}^{d_i}(t_k^i - \bar{t}_i)^2$.

For example, let $n=3$, $d_A=2$, $d_B=2$ and $d_C=4$; thus, $D=8$, $\bar{t}_A=4$, $\bar{t}_B=4$ and $\bar{t}_C=2.$ Any sequence is a feasible solution. For example, the sequence (C, A, C, B, C, B, A, C) is a solution, where $RTV=\left((5-4)^2+(3-4)^2\right)+\left((2-4)^2+(6-4)^2\right)$ $+\left((2-2)^2+(2-2)^2+(3-2)^2+(1-2)^2\right)=2+8+2=12$.

Corominas et al. (2007) presented the RTVP and proposed a mixed integer linear programming (MILP) model and five heuristic algorithms for solving it. García et al. (2006) presented six metaheuristic algorithms: a multi-start, a greedy randomized adaptive search procedure (GRASP) and four variants of a discrete particle swarm optimization (PSO) algorithm. Other ten discrete PSO algorithms were proposed in García-Villoria and Pastor (2009). A cross-entropy method approach was used in García-Villoria et al. (2007). The Electromagnetism-like Mechanism was proposed to solve the RTVP in García-Villoria and Pastor (2008a). Finally, the best heuristic results recorded to date have been obtained with a Psychoclonal algorithm (García-Villoria and Pastor, 2008b). Corominas et al. (2006) presented an improved MILP model; in our paper a slight simplification of this improved MILP model is

doi:10.3926/jiem.2008.v1n2.p4-15 ©© JIEM, 2008 – 01(02): 4-15 - ISSN: 2013-0953

considered. The model to be tested, which is presented in the Annex, consists of an objective function to be minimised, labelled (1), and 10 sets of constraints, labelled (2) to (11).

3 Computational experiment

Using the RTVP as the test problem, a computational experiment was performed in order to show that the order of the constraints in the model can have a considerable effect on the calculation time of the mathematical programming solver. The basic data used for the experiment are as follows:

- The 20 instances in Computational Experiment 1 in Corominas et al. (2006) were used. In short, D was between 21 and 40 units, n was between 3 and 12 products, and d_i was between 1 and 15 units.

- Sets (2) to (11), the 10 sets of constraints of the model shown in the Annex, were grouped into seven groups of constraints (G1 to G7). All possible permutations of these seven groups of constraints were carried out; therefore, 7! MILP models were generated and solved optimally (5,040 models). Table 1 shows the 10 original sets of constraints $(Original)$ and the seven groups of constraints that were used to test the order $(Groups)$. The grouping of the original sets of constraints was carried out in accordance with the similarity between the characteristics of the RTVP that were modeled by the restrictions. As explained later in this paper, grouping the constraints into seven groups shows that the order of the constraints in the model has a decisive influence on the efficiency; therefore, it was not necessary to generate and solve optimally the 10! MILP models (it would indeed be highly time-consuming: solving the 7! MILP models took 4,550,156 seconds).

- The MILP models were solved to optimality using the ILOG CPLEX 9.0 optimizer and a Pentium IV PC at 3.4 GHz with 1 GB of RAM. These models were run with the same computer environment in order to the performance (calculation time) of the next run was not affected.

doi:10.3926/jiem.2008.v1n2.p4-15 ©© JIEM, 2008 – 01(02): 4-15 - ISSN: 2013-0953

Original	Groups
(2)	G1
(3)	G2
(4)	
(5)	G3
(6)	G4
(7)	
(8)	G5
(9)	G6
(10)	G7
(11)	

Table 1. "Grouping of the original sets of constraints".

The 5,040 MILP models were generated and solved optimally for each of the 20 test instances (a total of 100,800 models). Obviously, the number of variables and constraints of the 5,040 models were the same for each instance, because the only difference between these MILP models was the order of the seven groups of constraints. For each of the 5,040 models the sum of the calculation time of the 20 instances was obtained. Table 2 shows the permutations required by the minimum and maximum calculation times (in seconds).

Permutation of the groups	Time (in seconds)
G5_G2_G3_G6_G1_G7_G4	335
G6_G5_G4_G2_G7_G3_G1	5.851

Table 2. "Best and worst permutations of the groups of constraints".

Table 2 shows the ILOG CPLEX 9.0 optimizer takes only 335 seconds for permutation G5_G2_G3_G6_G1_G7_G4, whereas for permutation G6_G5_G4_G2_G7_G3_G1 it takes 5,851 seconds. That is, in order to solve 20 instances of this test problem, the order of the constraints in the model returns a ratio of 17.47 between the maximum and the minimum calculation times of the ILOG CPLEX solver.

4 Conclusions

This paper presents a new, unexpected feature that should be taken into account whenever a mixed integer linear mathematical formulation must be solved optimally with a mathematical programming solver: the order of the constraints in the model can considerably change the calculation time of the solver.

doi:10.3926/jiem.2008.v1n2.p4-15 ©© JIEM, 2008 – 01(02): 4-15 - ISSN: 2013-0953

In order to reveal this difference in performance, a computational experiment was conducted in which the Response Time Variability Problem (RTVP) was solved using the ILOG CPLEX 9.0 optimizer. It returned a ratio of 17.47 between the maximum and the minimum calculation times necessary to solve optimally 20 instances of the RTVP, according to the order of the constraints in the model. With one permutation of the sets of constraints the solver needs only 335 seconds, whereas with another one it needs 5,851 seconds. It has thus been shown that the efficiency of the mixed integer linear mathematical programming depends not only on the model used, but also on how the information is introduced into the solver.

Because the ILOG CPLEX 9.0 optimizer is commercial software, it is very difficult for the authors to explain its performance. One hypothesis, which is not confirmed, is that the order in which the variables are handled in the search process depends on the order of the constraints in the model. Obviously, if we formalize the RTVP with another model (or we solve another problem) the difference in performance according to the order of the constraints in the model could be lower, but it could also be even higher.

Future research work will involve checking if this new, unexpected feature is also showed when another mathematical programming solver is used. The possibility to establish a formalised procedure to achieve, a priori, the best order of the constraints in the model is another area of future research.

Annex

The RTVP model to be tested is specifically as follows.

Data:

n Number of products $(i = 1,...,n)$

D Total number of units

d_i Number of units of product i $(i = 1,...,n)$ to be scheduled $(k = 1,...,d_i)$; it is assumed that $\sum_{i=1}^{n} d_i = D$

doi:10.3926/jiem.2008.v1n2.p4-15 ©© JIEM, 2008 – 01(02): 4-15 - ISSN: 2013-0953

$\overline{t_i}$ Average distance between two consecutive units of product i: $\overline{t_i} = D/d_i \ \ (i = 1,...,n)$

$G1$ Set of products with $d_i \geq 2$: $G1 = \{i \mid d_i \geq 2\}$

LB_i, UB_i Lower and upper bound on the distance between two consecutive units of product i $(\forall i \in G1)$

E_{ik}, L_{ik} First and last position that can be occupied by unit k of product i $(i = 1,...,n; k = 1,...,d_i)$

H_{ik} Set of positions that can be occupied by unit k of product i: $H_{ik} = \{h \mid E_{ik} \leq h \leq L_{ik}\}$ $(i = 1,...,n; k = 1,...,d_i)$

Variables:

sl_{ik} Position of unit k of product i $(\forall i \in G1; k = 1,...,d_i)$

$y_{ikh} \in \{0,1\}$ 1 if and only if unit k of product i is placed in position h ($i = 1,...,n$; $k = 1,...,d_i$; $h \in H_{ik}$)

$\gamma_{ik}^{j} \in \{0,1\}$ 1 if and only if the distance between units k and $k+1$ of product i is greater than or equal to $LB_i + j$ $(\forall i \in G1; k = 1,...,d_i; j = 1,...,UB_i - LB_i)$

Model:

$$[MIN]\ RTV = \sum_{\forall i \in G1, k, j} \left((LB_i + j)^2 - (LB_i + j - 1)^2 \right) \gamma_{ik}^{j} + \sum_{i \in G1} d_i \cdot LB_i^2 - \sum_{i \in G1} d_i \cdot \overline{t}^2 \qquad (1)$$

$$\sum_{h \in H_{ik}} h \cdot y_{ikh} = sl_{ik} \qquad (\forall i \in G1; k = 1,...,d_i) \qquad (2)$$

$$sl_{i,k+1} - sl_{i,k} = LB_i + \gamma_{ik}^{1} + ... + \gamma_{ik}^{j} + ... + \gamma_{ik}^{UB_i - LB_i} \qquad (\forall i \in G1; k = 1,...,d_i - 1) \qquad (3)$$

doi:10.3926/jiem.2008.v1n2.p4-15 ©© JIEM, 2008 – 01(02): 4-15 - ISSN: 2013-0953

$$D - sl_{i,d_i} + sl_{i,1} = LB_i + \gamma_{i,d_i}^{1} + \ldots + \gamma_{i,d_i}^{j} + \ldots + \gamma_{i,d_i}^{UB_i - LB_i} \qquad (\forall i \in G1) \qquad (4)$$

$$\gamma_{ik}^{j} \geq \gamma_{ik}^{j+1} \qquad (\forall i \in G1; k = 1,\ldots,d_i; j = 1,\ldots,UB_i - LB_i - 1) \qquad (5)$$

$$\sum_{\forall (i,k) | h \in H_{ik}} y_{ikh} = 1 \qquad (h = 1,\ldots,D) \qquad (6)$$

$$\sum_{h \in H_{ik}} y_{ikh} = 1 \qquad (i = 1,\ldots,n; k = 1,\ldots,d_i) \qquad (7)$$

$$\sum_{k=1}^{d_1 - 1} k^2 \cdot \left(sl_{1,k+1} - sl_{1k}\right) + d_1^{\,2} \cdot \left(D - sl_{1d_1} + sl_{11}\right) \leq$$

$$\sum_{k=1}^{d_1 - 1} \left(d_1 - k + 1\right)^2 \cdot \left(sl_{1,k+1} - sl_{1k}\right) + 1^2 \cdot \left(D - sl_{1d_1} + sl_{11}\right) \qquad (8)$$

$$\sum_{k=1}^{d_1 - 1} k^2 \cdot \left(sl_{1,k+1} - sl_{1k}\right) + d_1^{\,2} \cdot \left(D - sl_{1d_1} + sl_{11}\right) \leq$$

$$\sum_{k=1}^{d_1 - 1} \left(1 + \left(\left(k + j\right) \bmod d_1\right)\right)^2 \cdot \left(sl_{1,k+1} - sl_{1k}\right) + \left(1 + j\right)^2 \cdot \left(D - sl_{1d_1} + sl_{11}\right) \left(j = 0,\ldots,d_1 - 2\right) \qquad (9)$$

$$\sum_{j=1}^{UB_i - LB_i} \sum_{k=1}^{d_i} \gamma_{ik}^{j} = D - d_i \cdot LB_i \qquad (\forall i \in G1) \qquad (10)$$

$$\sum_{j=1}^{UB_i - LB_i} \sum_{k=1}^{d_i} \left(1 - \gamma_{ik}^{j}\right) = d_i \cdot UB_i - D \qquad (\forall i \in G1) \qquad (11)$$

Objective function (1) minimises the response time variability. (2) provides the position of unit k of product i $\left(sl_{ik}\right)$. (3) and (4) ensure that the distance between units k and $k+1$ of product i is equal to an integer value between 1 and UB_i. (4) refers to the distance between the first unit of product i in a cycle and the last unit of the same product in the preceding cycle. (5) imposes the consistency of values for γ_{ik}^{j}. (6) ensures that one and only one unit is placed in each position h of the sequence and constraint set (7) ensures that each unit k of each type of product i is assigned to one and only one position of the sequence. The first unit of one

doi:10.3926/jiem.2008.v1n2.p4-15 ©© JIEM, 2008 – 01(02): 4-15 - ISSN: 2013-0953

product i with the largest d_i is fixed in the first position of the sequence (we can assume, without loss of generality, that this product is product 1); then (8) calculates two values for the sequence, considering it first in clockwise order and then in anticlockwise order and it is imposed that the value for the former is less than or equal to the value for the latter. With (9) a value is calculated for the sequence by shifting the units of product i with the largest d_i (we can assume, without loss of generality, that this product is product 1); then, it is imposed that the value of the first sequence is less than or equal to the "value" of the other sequences. Finally, for each product $i \in G1$, it is imposed, over variables γ_{ik}^{j}, that the sum of the distances between its units is equal to D ((10) and (11)).

Acknowledgments

The author is very grateful to Professor Albert Corominas (Technical University of Catalonia) for his valuable comments, which have helped to enhance this paper. This research has been supported by the Spanish MCyT project DPI2007-61905 co-financed by the FEDER

References

Bar-Noy, A., Bhatia, R., Naor, J., & Schieber, B. (2002). Minimizing service and operation costs of periodic scheduling. *Mathematics of Operations Research*, 27, 518-544.

Billionnet, A. (1999). Integer programming to schedule a hierarchical workforce with variable demands. *European Journal of Operational Research*, 114, 105–114.

Bollapragada, S., Bussieck, M.R., Mallik, S. (2004). Scheduling commercial videotapes in broadcast television. *Operations Research*, 52(5), 679-689.

Brusco, M.J. (2008). Scheduling advertising slots for television. *Journal of the Operational Research Society*, 59, 1363-1372.

Corominas, A., Kubiak, W., Pastor, R. (2006). Solving the Response Time Variability Problem (RTVP) by means of mathematical programming. *Working paper IOC-DT*, Universistat Politècnica de Catalunya, Spain.

doi:10.3926/jiem.2008.v1n2.p4-15 ©© JIEM, 2008 – 01(02): 4-15 - ISSN: 2013-0953

Corominas, A., Pasto,r R., & Plans, J. (2008). Balancing assembly line with skilled and unskilled workers. *OMEGA The International Journal of Management Sciences*, 36, 1126–1132.

Corominas, A., Kubiak, W., & Moreno, N. (2007). Response time variability. *Journal of Scheduling*, 10, 97–110.

García, A., Pastor, R., & Corominas, A. (2006). Solving the Response Time Variability Problem by means of metaheuristics. *Frontiers in Artificial Intelligence and Applications*, 146, 187–194.

García-Villoria, A., Pastor, R., & Corominas, A. (2007). Solving the Response Time Variability Problem by means of the Cross-Entropy Method. *International Journal of Manufacturing Technology and Management* (to be published).

García-Villoria, A., & Pastor, R. (2008ª). Solving the Response Time Variability Problem by means of the Electromagnetism-like Mechanism. *Working paper IOC-DT-P-2008-03*, Universitat Politècnica de Catalunya, Spain (available at http://hdl.handle.net/2117/2013).

García-Villoria, A., & Pastor, R. (2008b). Solving the Response Time Variability Problem by means of a Psychoclonal Approach. *Journal of Heuristics*. doi:10.1007/s10732-008-9082-2.

García-Villoria, A., & Pastor, R. (2009). Introducing dynamic diversity into a discrete Particle Swarm Optimization. *Computers & Operations Research*, 36(3), 951-966.

Herrmann, J.W. (2007). Generating Cyclic Fair Sequences using Aggregation and Stride Scheduling. *Technical Report*, University of Maryland, USA.

Monden, Y. (1983). *Toyota Production Systems*. Industrial Engineering and Management Press: Norcross, GA.

Pastor, R., Altimiras, J., & Mateo, M. (2008). Planning production using mathematical programming: The case of a woodturning company. *Computers & Operations Research*. doi: 10.1016/j.cor.2008.08.005

doi:10.3926/jiem.2008.v1n2.p4-15 ©© JIEM, 2008 – 01(02): 4-15 - ISSN: 2013-0953

Margot, F. (2007). Symmetric ILP: Coloring and small integers. *Discrete Optimization*, 4, 40–62.

Salkin, H.M., & Mathur, K. 1989. *Foundations of integer programming*, North-Holland, Amsterdam.

Waldspurger, C.A., & Weihl, W.E. (1995). *Stride scheduling: deterministic proportional-share resource management*. Technical Memorandum MIT/LCS/TM-528. MIT, Laboratory for Computer Science, Cambridge.