

*MRP II, linear optimization,  
relational databases, ECL<sup>i</sup>PS<sup>e</sup>-CLP.*

Jarosław WIKAREK [0000-0003-4758-1490]\*,  
Paweł SITEK [0000-0001-6108-0241]\*, Mieczysław JAGODZIŃSKI\*\*

## A DECLARATIVE APPROACH TO SHOP ORDERS OPTIMIZATION

### Abstract

*The paper presents the problem of material requirements planning with optimization of load distribution between work centers and workers' groups. Moreover, it discusses the computational example for shop orders optimization. The data for this example were taken from the relational database. The method of Constraint Logic Programming (CLP) for shop orders optimization has been suggested. Using Constraint Logic Programming, the constraints may be directly introduced to the problem declaration, which is equivalent to the source code of the program. The ECL<sup>i</sup>PS<sup>e</sup>-CLP software system has been presented. It allows for solving optimization problems concerning dimensions greater than in the case of the professional mathematical programming solver "LINGO". The application of ECL<sup>i</sup>PS<sup>e</sup>-CLP in accessing data from relational databases has been presented.*

### 1. INTRODUCTION

Information systems have long been an important part of the manufacturing environment. In the 1960s, manufacturers developed Material Requirements Planning (MRP). According to the American Production and Inventory Control Society, Inc. (APICS), MRP is a set of techniques that uses bill of material data, inventory data, and the master production schedule to calculate requirements for materials. It makes recommendations for reordering materials. Furthermore, because it is time-phased, it makes recommendations for rescheduling open

---

\* Technical University of Kielce, Control and Management Systems Section, 1000-PP 7, Kielce, Poland, j.wikarek@tu.kielce.pl, sitek@tu.kielce.pl

\*\* Silesian University of Technology, Department of Automatics, Akademicka 16, 44-101 Gliwice, Poland, mieczyslaw.jagodzynski@polsl.pl

orders when due dates and need dates are not in phase. Time-phased MRP begins with the items being listed on the Master Production Schedule and the determination of the quantity of all components and materials required to fabricate those items, and the date that the components and materials are required. Time-phased MRP is accomplished by exploding the bill of material, adjusting for inventory quantities on hand or on order and offsetting the net requirements by the appropriate lead times.

In the late 1970s and early 1980s, manufacturers integrated MRP (Material Requirement Planning) and other manufacturing and business functions. This renaissance is commonly known as Manufacturing Resource Planning (MRP II). According to the American Production and Inventory Control Society, Inc. (APICS), MRP II is a method for the effective planning of all the resources of a manufacturing company. Ideally, it addresses operational planning in units, financial planning in currency, and has a simulation capability to answer “what if” questions. It is made up of a variety of functions, each linked together:

- business planning,
- sales and operation planning,
- demand planning,
- master scheduling,
- material planning,
- capacity planning,

and the execution support system for capacity and material. Output from these systems can be integrated with financial reports (Landvater & Gray, 1989). MRP II has also been defined as a set of software modules based on an integrated database. Within this paradigm, the main modules are MPS (Master Production Schedule), MRP (Material Requirements Planning), CRP (Capacity Requirements Planning) and SFC (Shop Flow Control). Decisions for each layer of the manufacturing system are generated and processed in these modules. In the MPS module, master production schedule orders are generated based on sales forecasts and information about real customer orders. In the MRP module, planned orders are generated based on master production schedule orders, inventory stock, bill of material, etc. The SFC module, together with the CRP module, generates shop orders on the ground of planned orders and information about work centers’ capacity, routings and real inventory stock. These decisions may be more or less detailed, according to their layers and areas of influence.

## 2. DECISIONS IN MRP II SYSTEMS

The MRP II standard does not assure optimum values of generated orders. What is more, their feasibility is not assured automatically. Before the realization of shop orders, the charge of work centers and of workers' groups, which depends on them, is checked in the CRP module. If the work centers' capacity is insufficient, then the planned orders are corrected in dialogue with the user of the computer system.

The user of the MRP II system changes the schedule of allowance of work centers regarding planned orders for as long as the CRP module signals insufficient production capacity. If this procedure is not successful, the user should change the planned orders that have been received previously from MRP module. Sometimes, changes of master production schedule orders, made after negotiations with customers, are inevitable.

## 3. SHOP ORDERS OPTIMIZATION

For automating and optimizing the decision procedure described in chapter 2, the mathematical model of material requirements planning with optimization of load distribution between work centers and workers' groups has been suggested. This model has been formulated as a linear programming problem. A detailed discussion of its objective function (1) and constraints has been presented in (Sitek, Wikarek & Zaborowski, 2002).

The main part of the goal function (1), which is minimized in the presented model of optimization of load distribution between work centers and workers' groups, is the cost of charge for work centers and workers' groups. A work center is a group of similar/identical machines, each of them capable of executing every shop task from a definite set of tasks allotted for a given center. A workers' group consists of workers with the same specializations and load cost per hour. Each of them could execute every shop task from a definite set of tasks allotted for a given group. The constraints of decision variables in this model are a formal representation of the constraints in the MRP II method. Therefore, every feasible solution of this model (constraints) represents the planned orders which have been executed without needs of later modification. Possible corrections in chance of contradicting the constraints are MPS orders. Therefore, one introduced to the goal function the corrective variable of the MPS order quantity  $X_j(\tau)$ . The penalties for the product  $j$  storage and above corrective variables of the MPS were additional elements of the goal function. These additional elements have to assure agreement among the others with the MRP II standard (Landvater & Gray, 1989).

The area of shop orders optimization has been shown in Figure 1.

The main decision variables of this model have been shown in Table 1. Other variables and indices are listed below:

- (j) – an inventory item,  $j \in J$ , ( $J \supset J^M$  – the set of products)
- (j,k) – an operation identifier,  $k \in K_j, j \in J$
- (j,r) – a routing identifier,  $r \in I_j^R, j \in J^M$
- (j,r,k) – a shop operation identifier
- h – a work center number,  $h \in I^{WC}$
- p – a workers group number
- q – a worker specialization number
- t – an operational planning period number,  $t \in T$
- $G_j(\tau)$  – the quantity of the MPS order (j,  $\tau$ )
- $\gamma_h$  – the work center load cost per hour,  $h \in I^{WC}$
- $\Omega_{jrk}(\tau)$  – machine hours of the shop operation (j,r,k) in the period  $\tau$
- $\delta_p$  – the workers group load cost per hour
- $\beta_j$  – the penalty for the MPS order correction of the product j
- $c_j$  – the penalty for the product j storage

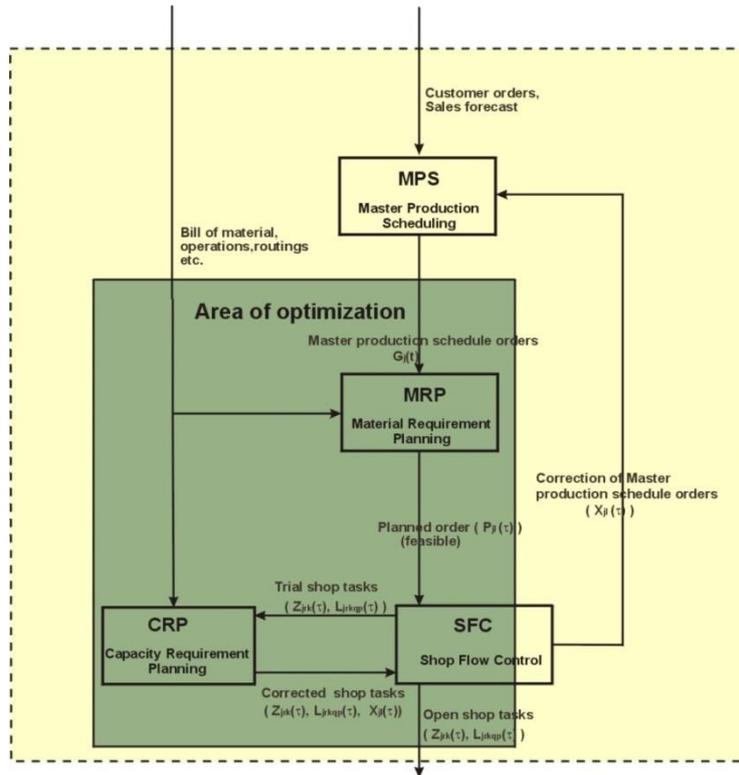


Fig. 1. The area of shop orders optimization in production control in a MRP II system

$$\sum_{\tau=1}^{\tau^h} \sum_{j,r \in J_j^h} \sum_{k \in K_j} \left\{ \gamma_h \Omega_{jrk}(\tau) + \sum_{q \in I_{jk}^A} \sum_{p \in I_q^L} \delta_p L_{jrkqp}(\tau) \right\} + \sum_{j \in J^Y} \sum_{t=1}^{\tau^h + L_j^{\Sigma}} \beta_j X_j(\tau) + \sum_{j \in J} \sum_{t=1}^{\tau^h} c_j V_j(\tau) \quad (1)$$

**Tab. 1. Decision variables of the optimization problem**

Symbol	Description
$P_j(\tau)$	Planned order quantity of the order $(j, \tau)$
$Z_{j,r,k}(\tau)$	Shop task quantity of the task $(j, r, k, \tau)$
$X_j(\tau)$	Corrective variable of the MPS order quantity of the order $(j, \tau)$
$L_{j,r,k,q,p}(\tau)$	Labor hours of workers group $p$ with specialization $q$ resulting from the shop task quantity $(Z_{j,r,k}(\tau))$ of the task $(j, r, k, \tau)$
$R_j(t)$	Requirement of the product $j$ in the period $t$
$V_j(t)$	Requested inventory of the product $j$ in the end of the period $t$ , (inventory $V_j(0)$ is given)

#### 4. COMPUTATIONAL METHODS FOR SHOP ORDERS OPTIMIZATION

Because the considered problem of shop orders optimization is a linear programming problem, the simplex method was first applied solve it (www.lindo.com, 2018). The software package “LINGO” was used to solve several optimization examples (www.lindo.com, 2018).

“LINGO” is a mathematical modeling language. Unlike conventional languages, such as Basic, Pascal or C, is nonprocedural. That is, when you specify a model for “LINGO” to solve, you only tell it what you want, not how it should find the solution. It is “LINGO’s” job to worry about the how. In this sense “LINGO” is known as a specification language. “LINGO’s” modeling language lets you express your problem in a natural manner which is very similar to standard mathematical notation.

This software is adequate for many examples. Unfortunately, it was not possible to find a solution for greater size problems.

Thus, it was necessary to examine an alternative method of optimization. A subsequent method, named the CLP (Constraint Logic Programming), has been applied. CLP may be defined as a body of techniques used for solving problems with constraints. The main idea of CLP is that (Rossi, Van Beek & Walsh, 2006; Niederliński, 2011):

- The problems to be solved are modeled using elementary logic, in a way that turns a model into a part of the problem-solving program.
- Exploration of the constraints, which should be satisfied by the solutions, generates solutions.

Using Constraint Logic Programming (CLP) for solving the optimization problem (Niederliński, 2011), its constraints, and the objective function (1) may be directly introduced to the problem declaration which is equivalent to the source code of the software program.

## 5. DATA STRUCTURE

The heart of most software implementations of the MRP II system is an integrated database. There are different database technologies available: hierarchical databases, network databases, relational databases and object-oriented databases. The relational database technology is, by virtue of being both the subject of international standards and a solid theoretical platform, by far the most widely spread database technology today. It is commercially available from a great number of vendors, like Oracle, Informix, Sybase etc. A relational database is organized in tables. A table represents a real world entity or concept, like a customer, a vendor, orders, invoices, machines etc. Each table is made up by a number of columns, some of which are used for data storage and others for keeping references to other tables. Together, the tables build a more or less complex structure, which will be referred to as the database structure. An exemplified database structure for the MRP II environment has been suggested in (Landvater & Gray 1989).

The data for the shop orders optimization were taken from relational databases. When using the CLP method or the “LINGO” software package for solving optimization problems, it was necessary to organize access to data (Figure 2). In the MRP II system, all data and decisions are stored in relational databases. When using the ECL<sup>i</sup>PS<sup>e</sup>-CLP ([www.eclipse.org](http://www.eclipse.org), 2018) language in the above environment, it was necessary to solve some problems. The most important were:

- Writing procedures for data export from a relation database to text files;
- Writing predicates for data import from text files to lists of coefficients in the ECL<sup>i</sup>PS<sup>e</sup>-CLP language;
- Implementing the optimization model to the source code of the software program in ECL<sup>i</sup>PS<sup>e</sup>-CLP;
- Starting optimization;
- Writing predicates for saving output data to text files;
- Writing procedures for data import from text files to a relation database.

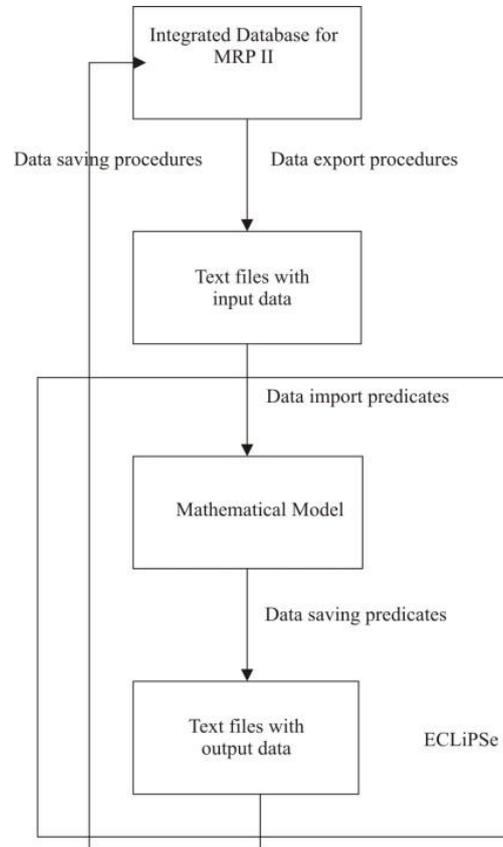


Fig. 2. Access to data for optimization shop orders in MRP II system

## 6. APPLICATION ECLiPSe-CLP TO SHOP ORDERS OPTIMIZATION

The implementation of the ECLiPSe-CLP (Niederliński, 2011; www.eclipse.org 2019) language in decision optimization has been shown in the example of shop orders optimization. One of the most important constraints of the above problem are material balance equations (2), defining net requirements in the given time period, which are equal gross requirements regarding inventory. Planned gross requirements are defined in every time period for every item.

$$V_j(\tau) = V_j(\tau-1) + P_i(\tau) - R_j(\tau) \text{ for } \tau = 1..T + L_j^z, j \in J, \quad (2)$$

The details of the implementation of the material balance constraint and access to data in the ECLIPSE-CLP language have been presented in the listings (from listing 1 to listing 7).

In addition, when implementing the above problem in ECLIPSE-CLP, it is possible to split the computation into several parts (computational levels). These parts have been shown in Table 2. Using ECLIPSE-CLP and splitting the optimization problem in parts have resulted in a more effective optimization and the possibility of solving problems of size greater than in the case of using the “LINGO” system.

```

% PREDICATE FOR MAKING LIST
% parameters
% 1 - the name of the list
% 2 - the end of previous list
% 3 - the size of the list,
% 4 - domain size,
li_t(L,L,0,0).
li_t([Zm|R],L,Nr,X):-
    Zm::0:X,
    Nr_1 is Nr-1,
    ( Nr_1#0 then
        li_t(R,L,Nr_1,0);
        li_t(R,L,Nr_1,X)).

```

**Fig. 3. Listing of predicate for making list**

```

L_j ::1..300,% number of items,
L_t ::1..20, % number of time periods
L_t1::1..20, % number of time periods
L_jt::1..6000,% product of items and time periods

```

**Fig. 4. Listing of declaration of domain sizes**

```

stale(L_j,L_t),
open('!dane\\Stale.txt',S,'r'),
read(S,Pomo), L_j # = Pomo,
read(S,Pomol),
L_t # = Pomol,
close(S).
czytaj(Plik,Lista,D):-
open(Plik,S,'r'),
li_c(Lista,[],D,S),
close(S).
li_c(L,L,0,0).
li_c([Zm|R],L,Nr,S):-
read(S,Pomo), Zm# = Pomo,
Nr_1 is Nr-1,
( Nr_1#0 then
    li_c(R,L,Nr_1,0);
    li_c(R,L,Nr_1,S)).

```

**Fig. 5. Listing of data import predicates**

```
og_6(Pjt,Rjt,Vjt,Vjt,L_j,L_t1),
```

**Fig. 6. Listing of call predicate for material balance constraint**

```
stale(L_j,L_t),
L_t1 is L_t+1,
L_jt is L_j*L_t1,
li_t(Vo,[],L_j,X),
czytaj('!dane\Vo.txt',Vo,L_j),
li_t(Rjt,[],L_j*L_t1,X),
li_t(Vjt,[],L_j*L_t1,X),
li_t(Pjt,[],L_j*L_t1,X),
zapas_0(Vjt),
```

**Fig. 7. Listing of call predicates**

```
% PREDICATE FOR MATERIAL BALANCE CONSTRAINT
%1 - Pjt - net requirements
%2 - Rjt - gross requirements,
%3 - Vjt - inventory of item,
%4 - [A|B] - inventory of item in the next period,
%5 - L_j - number of items,
%6 - L_t1 - numer of time periods,
og_6(Pjt,Rjt,Vjt,[A|B],L_j,L_t1):-
  og_6p(Pjt,Rjt,Vjt,B,L_j,L_t1,0,0).
og_6p([C|D],[E|F],[G|H],[I|J],L_j,L_t1,Ob,Cz):-
  Ob1 is Ob+1, Cz1 is Cz+1,
  (Cz1 #< L_t1 then
  C is E + I - G,
  ( C #>0 then
  I is 0;
  (C is 0, E is 0)),
  ( Cz1 #= L_t1 then
  Cz2 is 0;
  Cz2 is Cz1, Pomo is L_j*L_t1-1),
  (Ob1 #< Pomo then
  og_6p(D,F,H,J,L_j,L_t1,Ob1,Cz2);
  og_6p(D))).
og_6p([A|B]):-
  A is 0.
```

**Fig. 8. Listing of predicate for material balance constraint**

**Tab. 2. Computational levels**

Level	Solving sub-problem
1	Set to zero initial values of corrective variables, $X_j(\tau)$ , of MPS order quantities.
2	Calculate gross requirements $R_j(\tau)$ , planned order quantities $P_j(\tau)$ and planned inventory values $V_j(\tau)$ .
3	Distribute planned orders $P_j(\tau)$ between particular work centers and workers groups (calculate variables $Z_{jkw}(\tau)$ , $L_{jkwq}(\tau)$ ). If work centers load exceeds their capacity then go back to the step 2 with nonzero values of corrective variables $X_j(\tau)$ .

```

% PREDICATE FOR REQUESTED
% INVENTORY OF THE ITEM J IN
% THE END OF THE FIRST PERIOD
% 1 - [Vjt] - the list of inventory
%     item,
% 2 - [A|B] - the list of inventory
%     item,
% 3 - L_j - number of items,
% 4 - L_t - number of time periods,
% 5 - Obe - number of current
%     processing item,
zapas_0(Vjt,[A|B],L_j,L_t,Obe):-
  Ob1 is Obe+1,zap_pom(Vjt,A,L_j,L_t,Ob1,0,1,0),
  ( Ob1 #< L_j then
    zapas_0(Vjt,B,L_j,L_t,Ob1)).

```

**Fig. 9. Listing of predicate for requested inventory**

## 7. NUMERICAL EXPERIMENTS

Numerous computational experiments were performed to verify the model and the proposed approach. Individual experiments differed in the number of products (N from 5 to 40) performed as part of the order. Two possible execution routes were adopted for each product, which contained from 3 to 5 operations. The availability of 20 employees with 6 specialties was also assumed.

For this data, the proposed problem was modeled and solved in two environments, i.e. the classic mathematical programming environment – LINGO – and in a declarative environment, constraint logic programming – ECL'PS<sup>e</sup>-CLP. The results are presented in Table 3. The advantage of using the declarative environment is clearly visible due to the time of calculations.

**Tab. 3. Computational levels**

N	LINGO		ECLiPS <sup>e</sup> -CLP	
	Time [s]	Fc	Time [s]	Fc
5	25	234	10	234
10	56	456	14	456
20	145	934	24	934
30	345	1435	34	1435
40	600	NFSF	45	2034

## 8. CONCLUSION

The results of calculations for numerical experiments, obtained by using the “LINGO” system and the ECLiPS<sup>e</sup>-CLP language, are the same. Other examples proved that using an ECLiPS<sup>e</sup>-CLP language software system enables solving optimization problems of dimensions greater than in the case of “LINGO”. The application of the ECLiPS<sup>e</sup>-CLP language to optimization and also to access data from relational databases proved to be an interesting solution. In the future, to model and solve this problem, it is planned to use a proprietary hybrid approach that integrates both of these environments (Sitek & Wikarek, 2018, 2019).

## REFERENCES

- Landvater, D. V., & Gray, C. D. (1989). *MRP II Standard System*. Oliver Wight Publications.
- Niederliński, A. (2011). *A Quick and Gentle Guide to Constraint Logic Programming via ECLiPSe*. Gliwice.
- Rossi, F., Van Beek, P., & Walsh, T. (2006). *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. New York, NY, USA: Elsevier Science Inc.
- Sitek, P., & Wikarek, J. (2019). Capacitated vehicle routing problem with pick-up and alternative delivery (CVRPPAD): model and implementation using hybrid approach. *Annals of Operations Research*, 273, 257. doi:10.1007/s10479-017-2722-x
- Sitek, P., & Wikarek, J. (2018). A multi-level approach to ubiquitous modeling and solving constraints in combinatorial optimization problems in production and distribution. *J. Appl. Intell*, 48, 1344. doi:10.1007/s10489-017-1107-9
- Sitek, P., Wikarek, J., & Zaborowski, M. (2002). Application of Constraint Logic Programming (CLP) to Shop Orders Optimization in MRP II Systems. In *Conference: 4th Workshop on Constrained Programming for Decision and Control* (pp. 59-66). Gliwice.
- [www.eclipse.org](http://www.eclipse.org) (n.d.) Retrieved October 19, 2018, from The Eclipse Foundation open source community website [www.eclipse.org](http://www.eclipse.org).
- [www.lindo.com](http://www.lindo.com) (n.d.) Retrieved October 19, 2018, from [www.lindo.com](http://www.lindo.com)