CORE DISCUSSION PAPER
2001/56

A GENERAL HEURISTIC FOR PRODUCTION PLANNING PROBLEMS

Mathieu VAN VYVE[1] and Yves POCHET[2]

December 2001

Abstract

We consider production planning problems with the restriction that all integer variables model setups. Since finding a feasible solution of such problems is in general NP-complete, the classical approaches have been the use of heuristics to find good feasible solutions on the one hand, or Branch&Cut on the other hand. In the case of the former, a dual bound is not available, and there is no guarantee of solution quality. For the latter, the accent has been on improving the dual bound and only the simplest schemes have been used to find good feasible solutions.

Here we first show that such simple schemes may run into trouble, even when applied to very simple problems. This motivates the proposed heuristic, IPE, which is designed to be used within a Branch&Cut approach. We test the performance of the heuristic on various published lotsizing and network design problems, with and without tightened reformulations. We compare these results with other heuristics and with time truncated B&B searches. IPE appears to be the best choice for large problems with weak formulations.

Keywords: production planning, heuristic, Branch&Cut, setup times, multilevel.

[1]Center of Operations Research and Econometrics (CORE), Université catholique de Louvain. 34, voie du Roman Pays. 1348 Louvain-la-Neuve, Belgium

[2]Center of Operations Research and Econometrics (CORE) and Institut d'Administration et de Gestion (IAG), Université catholique de Louvain. 34, voie du Roman Pays. 1348 Louvain-la-Neuve, Belgium

This text presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. The research was carried out with financial support of the project TMR-DONET ERB FMX-CT98-0202 of the European Community, and with support of the Growth Project G1RD-1999-00034 (LISCOS) of the European Community. The scientific responsibility is assumed by the authors.

1 Introduction

The aim of this paper is to present a heuristic for a restricted class of capacitated multi-item multi-level lot sizing problems with setup times. This problem has recently received much attention both because of its applicability to real life production planning problems and its complexity. In 1991, Maes et al.[21] have shown that setup times make the search for a feasible solution NP-Complete, while Arkin et al.[1] have shown the general uncapacitated multi-level problem to be NP-Hard. Hence, one cannot reasonably ask for a provably optimal solution of reasonable-size instances of such problems. However, its applicability has motivated a lot of research.

Two main approaches can be outlined. The first one is the design of heuristic schemes which mainly try to find good feasible solutions. As examples, we list here some recent contributions of this type. For a more complete survey of the subject, see for instance Kuik et al.[20]. Tempelmeier and Derstroff [30] have used a Lagrangean relaxation approach followed by a shifting production procedure to restore feasibility. Katok et al.[16] solve a series of linear programs without setup variables, modifying both objective and matrix coefficients to take setups implicitly into account. They then try to shift production to improve their solution by a restricted local search in the space of setup variables. Kim and Pardalos [17] have used a similar procedure for fixed charge network flow problems, but waiting for convergence. Simpson and Erenguc [29] have argued that, in relaxation approaches, the structure of the relaxed problem is too different from that of the original one to produce near optimal solutions. They instead use a modified lot-for-lot solution which is improved by local search heuristics. Meyer [23] uses a simulated annealing approach in solving multi-item multi-machine lot sizing problems. Despite the good results obtained, this heuristic approach suffers from a lack of informative lower bound, leaving the user without any measure of the quality of the solution. This is not true in general for Lagrangean approaches, but for the type of lotsizing problems considered here, the bounds obtained in this way are typically very weak.

The second approach is Branch and Cut (B&C). Indeed, adding cutting planes has proven to be a successful way to improve the formulation of lot sizing problems. Because of the tightness of the bounds provided, a branch and bound (B&B) tree may be used and truncated if needed. Pochet and Wolsey [27] show successful results in solving capacitated multi-item multi-level problems without setup times. Constantino [8, 9] extends these results to startups and lower bounds on production, while Miller [24, 25] generalizes to problems involving setup times. Many of those studies have been implemented in BC-PROD, a B&C code for solving lot sizing problems [4]. In [3], Belvaux and Wolsey show excellent computational results on various practical problems. It should be noted that in [3], much emphasis is placed on improving the lower bounds by adding strong valid inequalities, and much less on finding good feasible solutions during the branching process. Indeed, the search for primal solutions is embedded in the branching and node selection rules of the enumeration. In their simplest forms those rules rely on the fractional values of the LP solution. More complicated schemes using merit functions have also been used. However, the merit functions only take into account the change of the dual bound, see Cordier [10], which is logically independent of finding good feasible solutions.

The heuristic presented in this paper draws on both approaches. It is not an enumeration procedure, even though it can take advantage of cutting planes generated and branchings already performed, and is therefore well suited to be used within a B&C enumeration tree. It is based on the linear programming relaxation of the problem, and its main idea is to exploit the specific dependance between the setup variable and the associated production variable. There are two points that we particularly want to highlight in this paper. Firstly, the search for good feasible solutions during the enumeration process does not have to be done exclusively through specific branching or node selection rules. Secondly, strong cutting planes, designed to improve lower bounds, can be effectively used by primal heuristics (that are not based on progressively fixing integer variables).

The paper is organized as follows. In Section 2 we describe the generic MIP that we want to solve. In Section 3 we show that primal heuristics based on the value of the fractional solution may face a major difficulty. We then describe the proposed scheme, IPE, in Section 4. Finally, in Section 5 we describe a variety of lotsizing and network design problems, and in Section 6 evaluate the performance of our heuristic in solving these problems. We conclude in Section 7 with some directions for future research.

2 The model

The MIP problems we want to tackle have the following structure:

$$\begin{aligned} \min \quad & px + fy + hs && (1) \\ s.c. \quad & A_1 x + A_2 s = d && (2) \\ & B_1 x + B_2 y \leq D && (3) \\ & x - Cy \leq 0 && (4) \\ & y \in \{0,1\}^{|I|}, x, s \geq 0 && (5) \end{aligned}$$

with $B_1, B_2, C \geq 0$ and $C = diag(c_1, c_2, \cdots, c_{|I|})$, where I is the index set over which variables x and y are defined.

This model is fairly general and the following problems may be modelled by such a MIP :

Lot Sizing The x variables here represent production, the y variables stand for the setup associated with the production variables x and are used to model both setup costs and times. The s variables represent stocks and backlogs. Constraints (2) are typically inventory or flow conservation constraints, and (3) model capacity restrictions. The following extensions of the basic lot-sizing problem match the structure of the model : varying capacities, setup times, multi-item, multi-machine, multi-level, setup cost/time on stock. However, variable lower bounds on production, requiring additional constraints of the form $x - Ly \geq 0$, or start-ups requiring additional binary variables defined using different constraints other than (4), are not part of our generic model (1)-(5).

Network Design Many combinatorial problems that can be modelled as flow problems in a graph fit the above model perfectly. In this case, the x's are flow variables associated to the flow on the arcs of the graph, and the y's indicate whether an arc is opened or not. The s variables, if used, may represent flows without opening cost and therefore without associated binary variables. Constraints (2) enforce flow conservation at each node, constraints (4) model setup and capacity restrictions on the flows, but there are usually no constraints of type (3).

To refer to the lotsizing application, in the sequel, x variables will be called production variables and y variables will be called setup variables.

3 Motivation: a difficulty for LP-based heuristics

For LP-based heuristics, the idea of "rounding" the linear relaxation solution to an integer solution is natural. Such rounding heuristics fit into the following framework, where LP stands for the linear relaxation of problem (1)-(5).

Algorithm 1

1. *a) Let* $I_0 = I$
   $j = 0$
   b) Solve LP

2. *Until all* y *variables are integer :*
   a) $j = j + 1$
   b) Select a subset I_j *of* $I \setminus (\cup_{k=1}^{j-1} I_k)$
   c) For all $i \in I_j$*, select* $y'_i \in \{0, 1\}$ *and add the constraint* $y_i = y'_i$ *to* LP
   d) Solve LP
   e) if LP *is infeasible, backtrack*

The key operations in this algorithm are the selection (2.b) of the variables to be fixed, the choice (2.c) of the value at which those binary variables are to be fixed, and the backtracking operation (2.e). We describe how classical heuristics fit into this framework:

truncated B&B In a pure B&B search, the sets I_j are singletons, and the backtracking is always as limited as possible. The succession of 2.b and 2.c is referred to as "branching rule". In a truncated B&B we stop as soon as we have found a solution. In other words, algorithm 1 may be seen as a truncated B&B search with a depth first node selection strategy, and in which we dive $|I_j|$ levels at a time.

successive rounding Probably the most basic choice for the step 2.c is to round the variables to their nearest integer values. The selection of the variables in I_j at step 2.b may be predefined (for example, lexicographic order) or computed at each step (for example, based on the most (least) fractional variables in the current solution, as in [10]). When the sets I_j contain more than one variable, new values must be determined for y'_i when backtracking.

probabilistic fixing Another idea is to interpret the fractional value of variable y as the probability that y will be one at the optimal integer solution.

In that case, the set I_1 can be taken equal to I (all binary variables are fixed at the same time), and each variable y is fixed to one with a probability y_i^{LP}, the current fractional value of y_i. Several trials may be performed, until one or several feasible solution are found, or until a maximum number of iteration is reached, as in [7].

successive probabilistic fixing The two previous approaches may be mixed in the following way : the basic algorithm is as successive rounding, except that variables are not rounded but fixed according to a Bernouilli trial with probability y^{LP}. Note that this provides an elegant answer to the problem of determining new values y_i' in the case of backtracking.

Relax& Fix This heuristic has been used for lot-sizing problems, and is also referred to as "time decomposition heuristic", see [4]. The time interval $[1, NT]$ (corresponding to a planning horizon from period 1 up to period T) is divided in $Q = \lceil \frac{NT}{P} \rceil$ subintervals of length P. At step $2.b$, we choose I_j such that it contains all production variables in the time interval $[(j-1)P+1, jP]$. At step $2.c$, the values y_i' are computed as the optimal integer solution of the problem with variables contained in I_j treated as integer, variables in $\cup_{k=1}^{j-1} I_k$ are fixed and variables in $I \setminus (\cup_{k=1}^{j-1} I_k)$ relaxed to be continuous in $[0,1]$. P is typically choosen small enough such that these problems are easy to solve. This decomposition idea can be used for any problem where there exists a "natural" ordering of the binary variables.

Obviously, the larger P, the better the solution, but the running time increases exponentially with the size of P. A possible variant designed to improve the quality of the solution without risking exponential explosion is to use overlapping intervals.

All heuristics presented up to now, except Relax&Fix, use the value of the optimal LP solution y^{LP} to compute the value y'. But the fractional LP solution is not always a good indicator of the optimal MIP solution. This is illustrated by the following example.

Example 1

$$\min z = 5x_1 + 6x_2 + 30y_1 + 35y_2$$

$$x_1 + x_2 \geq 20$$

$$x_1 \leq 20y_1$$

$$x_2 \leq 100y_2$$

$$x \geq 0, y \in \{0,1\}$$

The optimal solution is

$$x_1 = 20, y_1 = 1, x_2 = 0, y_2 = 0$$

with $z = 130$*, but the LP solution is*

$$x_1 = 0, y_1 = 0, x_2 = 20, y_2 = 0.2$$

with $z = 127$*.*

In the example, the LP solution indicates the wrong choice for step 2.c. Moreover, the probabilistic heuristics will never find the optimal solution since y_1 has probability zero of taking the value one.

A particular feature of model (1)-(5) is crucial here : variables y are always taken as small as possible given the x variables. In the model, the only lower bound on y are the VUB constraints (4), and these are always active in the optimal solution. Hence, in model (1)-(5), and in Example 1, we always have $y_i^{LP} = x_i^{LP}/C$, and the solution is fractional whenever x_i^{LP} is nonzero and smaller than C. The values y_i^{LP} are underestimated unless $x_i^{LP} = 0$.

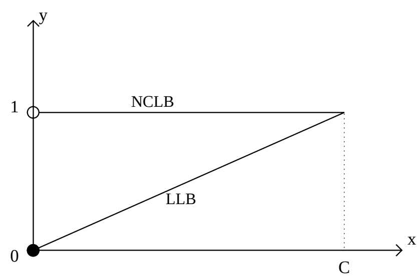
Observation 1 *In the solution of the linear relaxation of model (1)-(5), when x_i^{LP} is positive, it is impossible to distinguish the case where x_i^{LP} will remain positive even if the true setup cost and time are charged ; from the case where x_i^{LP} has to be zero because the setup is to expensive or takes too much time.*

Therefore, the value of the linear relaxation solution is not as such a good indicator of the integer optimal solution. The idea which will be explored in the next section is to change the model so that the new LP solution is a better indicator of a good integer solution.

4 The Iterative Production Estimate Heuristic

4.1 The scheme

When solving the linear relaxation of model (1)-(5), the Variable Upper Bound constraint (4) of the MIP model, which defines a noncontinuous lower bound (NCLB) on variable y, is approximated by a linear lower bound function (LLB) (see Figure 1).



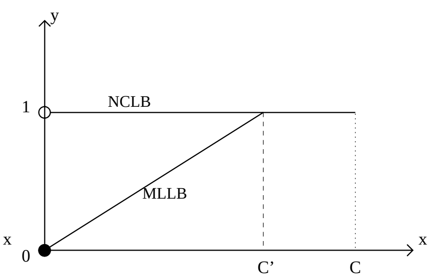


Figure 1: x-y relations in modified LP. Figure 2:x-y relations in IP and LP.

As a consequence, the setup variable is underestimated all over the domain. This is because we want the LP to be linear and to be a relaxation of the original problem, yielding a dual bound. But maybe this is not the best approximation to find a primal bound.

For example, if one has a good reason to think that production will be close to zero or C' in the optimal solution, one could use the modified linear lower bound (MLLB) approximation of Figure 2 to fix a lower bound on the setup variable. The associated setup variable will be close to zero or one in the new LP solution.

The basic idea of our scheme is thus to drop the requirement that we work with a relaxation of the initial model, and to use linear approximations allowing one to obtain fractional solutions which are closer to good integer solutions.

A natural candidate for C', an estimate of the level of production in the case it will be nonzero, is given by the LP solution. This leads to the following iterative algorithm, where $LP(\Gamma)$ denotes the linear relaxation of problem (1)-(5) using the vector Γ to define the VUB constraints (4) (i.e. $x \leq \Gamma y$).

Algorithm 2

1. $C' = C$
   solve $LP(C')$, *with solution* (x^{LP}, y^{LP}, s^{LP}).

2. *until all* y_i *variables,* $i \in I$, *are integer*
   a) For all $i \in I$ *such that* $y_i^{LP} \notin \{0,1\}$, *modify* C'_i *as* $C'_i = x_i^{LP}$
   b) solve $LP(C')$, *with solution* (x^{LP}, y^{LP}, s^{LP}).

There is however a big problem with this scheme : the production variables are monotonically decreasing because x_i cannot exceed C'_i. As a consequence, if at one iteration C'_i goes below the unknown optimal value x_i^*, the scheme will never reach the optimal solution. Two things may be done to overcome this drawback.

- In step $2.a$, substitute C'_i by $\lambda x_i^{LP} + (1-\lambda)C'_i$. This smoothing modification prevents jumping too fast.

- Remove the upper bound on y_i so that production x_i may increase beyond C'_i. And to prevent production to increase beyond the true capacity , we add an upper bound C_i on the production variables.

We obtain the final Iterative Production Estimate (IPE) heuristic, where $LPC(\Gamma)$ denotes the linear relaxation of problem (1)-(5), augmented with the upper bounds $x_i \leq C_i$, for all $i \in I$, without upper bounds on variables y_i, for all $i \in I$, and using the vector Γ to define the VUB constraints (4) (i.e. $x \leq \Gamma y$).

Algorithm 3 (IPE)

1. $C' = C$
   solve $LPC(C')$, *with solution* (x^{LP}, y^{LP}, s^{LP}).

2. *until all* y_i *variables,* $i \in I$, *are integer*
   a) For all $i \in I$ *such that* $y_i^{LP} \notin \{0,1\}$, *substitute* C'_i *by* $\lambda x_i^{LP} + (1-\lambda)C'_i$
   b) solve $LPC(C')$, *with solution* (x^{LP}, y^{LP}, s^{LP}).

3. *a) restore original problem (1)-(5)*
   b) solve the LP problem obtained by fixing the binary variables at their final values obtained at step 3.

The update of the VUB capacities C' in the new approximate linear formulation $LPC(C')$ tries to distinguish between the two cases mentioned in Observation 1.

If a variable y_i is fractional and less than one, C'_i will decrease at the next iteration, and the same production level x_i will induce a higher value of the associated setup variable y_i. So, the updated model tries to charge a higher setup time and cost to the production level x_i^{LP}, in order to observe whether the production level is maintained at this higher price, or production decreases because the new setup cost/time is too expensive.

Note that if we apply algorithm 3 to example 1, it converges to the optimal solution in two iterations.

4.2 Initialization

The scheme is initialized with $C' = C$. This ensures that the procedure starts with a relaxation of the initial problem. Therefore, infeasible linear relaxations can still be detected.

4.3 Stopping criterion

The stopping criterion supposes that the scheme converges (i.e. the setup variables converge to binary values). However, there is a priori no reason why this should be the case. But there is convergence in practice, and we leave the practical discussion to section 6.

In case of convergence, the obtained solution is feasible for the original problem, because:

- All binary variables are zero or one.
- All constraints except VUBs have been kept in the model.
- The production is bounded by C.
- There is no production if the corresponding setup variable is zero.

Therefore, IPE gives feasible solutions for any additional constraints, provided of course that it converges.

4.4 Setup reduction subroutine

In specific cases, the solution provided by IPE can be easily improved by eliminating setups. This is implemented as an option in IPE. More specifically, all non zero setup variables are sequentially set to zero and the corresponding LP solved. If a better solution is obtained, it becomes the incumbent. This has been used for the Trigeiro test problems (see Section 5.1).

4.5 Links with other schemes

The scheme has common features with that proposed in Harrison et al. [14], and generalised in Katok et al. [16], called CMSB. They also use a linear approximation of the dependance of the setup on production. But the spirit of their algorithm is different since they do not wait for the algorithm to converge, but rather use the solution obtained by running the algorithm for a couple of iterations as a starting point of a search heuristic in the space of the binary variables. This in turn implies a second difference, because the local search in the space of the binary variables gives the same results with or without cuts. The presence of cuts will affect the behaviour of their algorithm only in the first stage of the heuristic.

Kim and Pardalos [17] have used a similar approach (called DSSP) for fixed charge network flow problems. Instead of modifying the capacities in the VUB constraints, they get rid of the setup variables and treat them implicitly by updating the objective function coefficients of the production variables. For problems without setup times, the two algorithms, DSSP (with initialization of type II and updating scheme of type II) and IPE, are equivalent, except that DSSP checks if intermediate (i.e. fractional) solutions are not by chance good feasible solutions. In this sense, IPE can be considered as the extension of DSSP allowing one to tackle problems with setup times, or more general constraints of type (3).

5 The test set

The heuristic has been primarily designed for lot sizing problems but, as mentionned in Section 2, can be used for various combinatorial network problems. The test set will thus contain problems from both fields.

5.1 Lot Sizing Problems

The lot sizing test set contains problems from LOTSIZELIB [2] compatible with our generic model (1)-(5), i.e. those which do not contain variable lower bounds on production or start-up variables. A detailed description of each one can be found at the URL mentioned as reference. We have also added problems from Simpson and Erenguc [29]. They are multi-level capacitated problems with setup times and costs. Capacities, setup times and setup costs apply to families of products, which may have common products. Two different industrial applications (A and B), for which two data sets (1 and 2) and two different formulations (normal stock and echelon stock, see [6], [28]) have been used, yielding 2*2*2=8 models for 4 different problems (i.e. two models for each problem). They will be called SimEren.

For each problem, three formulations have been used. The first one is a straightforward instance of the general model (1)-(5). The second one is the same augmented with the cuts generated by XPRESS-MP [12] with default parameters. The third one is the original model augmented with the cuts generated by BC-PROD [3], a specialized code for lotsizing problems. For all the problems considered, cuts from BC-PROD were stronger than those from XPRESS-MP. Since we were not able to use BC-PROD with the echelon stock reformulations, there are only two reformulations for each SimEren problem with the echelon stock formulation.

Each model is also classified as easy (E), medium (M) or Difficult (D). The concept of "difficulty" of a MIP is elusive, because it is fuzzy and depends on many different criteria, such as the depth of the search tree, the number of feasible solutions, the branching rules used, etc. However, because of its usefulness for the analysis in the next section, such a classification has been made, according to the following rules :

- If two out of the three branching heuristics (see section 6.1) cannot find within five minutes a better solution than 130% of the best solution known, the problem is classified as Difficult.
- Models for which the optimal solution is found by at least one of the branching heuristics within five minutes are classified as Easy.
- All other problems are classified as Medium.

Appendix A.1 summarizes the important features of the different lotsizing problems.

5.2 Network problems

Various problems may be formulated as fixed charge network flow (FCNF) models. We consider single-commodity, single- and multi-source problems, with and without capacities. All problems are from [26], where a detailed account of the way they have been generated can be found. Here follow the important characteristics:

- Steiner tree problems may be modelled by a single source uncapacitated FCNF model with flow (variable) costs set to zero. One terminal node is arbitrarily selected as the source, while the others are sinks. The problems are available at $ftp://ftp.zib.de/pub/Packages/mp{-}testdata/index.html$, and a complete description of them can be found in [19].
- fixnet6 is from MIPLIB [5].
- [13] Multi-segment graphs are FCNFs with concave piecewise linear costs. They are modelled by duplicating arcs as many times as the number of segments in the piecewise linear function and by adding the constraint that at most one arc can have a positive flow. Our test problems have at most four segments per arc.
- Various kinds of graphs have been randomly generated using a modified NETGEN generator(see [18] for the original generator and [26] for the modifications). The names indicate the type and the size of the graphs: g stands for grid, k for complete, sp for series-parallel, r for random structure, p for planar. The two numbers indicate the number of nodes and arcs respectively. The structure of the series-parallel and planar graphs are determined using specific routines from LEDA [22]. For random graphs, arcs are created by randomly selecting two nodes as many times as the number of desired arcs.
- Problems h are from [15]. These are complete directed graphs with Euclidian distances as variable costs, plus one demand node linked to all the other nodes. Fixed costs are 50 times the variable ones.
- Capacitated graphs were built on the basis of uncapacitated graphs with a similar name. The capacities have been generated using the C/C++ random number generator.

Appendix A.2 summarizes the important features of the network problems.

6 Computational results

In this section, we would like to answer to the two following questions:

- How does the performance of IPE compare with that of other heuristics?
- What is the influence of cuts on the performance of IPE?

All algorithms have been coded in C with the help of the EMOSL and XOSL libraries of XPRESS [11]. To solve the linear programs, the primal simplex of XPRESS-MP version 11.50O has been used. All runs have been carried out on a 350Mhz PC machine with 128MB of RAM, running under Windows NT 4. Appendix B contains the complete computational results.

6.1 Lotsizing problems

We compared IPE with seven other heuristics:

BBdepth This is the standard XPRESS B&B with depth first search, truncated after 300 seconds. This corresponds to setting the XPRESS control parameters ndsel1 equal to 3.

BBbound This B&B strategy iteratively selects the open node with the best bound, then dives, selecting at each level the descendant node which has the best bound. This correponds to setting the XPRESS control parameters ndsel1 and ndsel2 equal to 1 and 3 respectively. The search is stopped after a maximum running time of 300 seconds.

BBestimate This is the same as BBbound, but it selects the node with the best estimate. This correponds to setting the XPRESS control parameters ndsel1 and ndsel2 equal to 1 and 2 respectively, with a maximum time of 300 seconds.

Relax&Fix It is implemented exactly as described in section 3, with Q equal to the number of periods.

CMSB This is the first part of the heuristic proposed in [16].

SSR This is CMSB, followed by the Simple Setup Reduction subroutine (see [16]).

RR This is SSR, followed by the Reduced Setup Reduction and Reduced Inventory Reduction subroutines (see [16]). RR corresponds to the full heuristic proposed in [16].

The three first heuristics, BBdepth, BBbound and BBestimate, will be referred as branching heuristics in the sequel.

6.1.1 IPE versus other heuristics

Table 1 and 2 summarize the performance of the eight heuristics on all problems, in terms of the quality of the solution found and the times respectively. The value of the solutions is expressed as a percentage of the best solution known for each problem, and the times as a percentage of IPE time. For the branching heuristics, the value of the solution found and the time taken are not well defined, because these heuristics do not stop unless they prove optimality or they reach the time limit. Because we want here to assess the performance of IPE, we define these concepts as follows: the value corresponds to the best solution found at the moment IPE finishes, while the time taken by the branching heuristics is the time taken to find a solution at least as good as the one found by IPE. For the other heuristics, the value of the solution found and the time taken are well-defined.

Because for some problems no value is available (when no solution has been found, the value is considered as infinite), to simply average those values does not make sense. Hence two ways of comparaison are presented : partial average and ranking. In partial average, we only take into account those values which are not more than 1000%. In ranking, we rank the heuristics from 1 to 8 for each problem by their performance (a high value indicates a good performance). Those ranking are then averaged, taking into account all models. The detailed results for all test problems can be found in Appendix.

Considering the quality of the solution found (see Table 1), IPE appears to perform best for both used criteria (partial average and rank). It should also be noted that IPE is the only one to have found a solution to all problems. Relax&Fix had difficulties with the SimpEren problem augmented with the XPRESS-MP cuts, while CMSB did not succeed to generate a feasible solution for one model.

The results concerning the times are displayed at Table 2. The situation is clear : the basic CMSB outperforms all others, but the price to pay is the poorest quality. The basic CMSB is thus eliminated. Since the clear second for time is IPE, both in terms of partial average and rank, the superiority of IPE over other heuristics as a default algorithm to find feasible solutions is justified.

Algorithm	BBestim	BBbound	BBdepth	IPE	RR	SSR	CMSB	Relax&Fix
partial average	129.6%	113.6%	125.1%	109.2%	115.7%	123.3%	142.0%	109.4%
rankings	3.72	4.10	3.03	6.40	4.62	3.41	2.34	6.10
% with no solutions	79%	72%	69%	100%	97%	97%	97%	88%

Table 1: Global quality performance

Algorithm	BBestim	BBbound	BBdepth	IPE	RR	SSR	CMSB	Relax&Fix
partial average	213.5%	245.2%	179.9%	100.0%	390.7%	387.0%	43.3%	179.2%
rankings	3.57	3.72	4.41	5.76	2.00	3.45	7.47	4.36

Table 2: Global time performance

However, a finer analysis is obtained if one looks at the results for each of the three groups (E, M and D) of problems seperately. Tables 3 summarizes those results. Broadly, BB search strategies perform well for easier problems. Their performance degrades on more difficult ones. Relax&Fix performs very well on easy and medium problems. On the contrary, the absolute performance of IPE does not deteriorate with the difficulty of the problem, which translates in a better rating compared to the other heuristics. IPE is less sensible to exponential

Figure 2: Number of iterations for problems of various sizes

explosion. This fact is illustrated in Figure 2, which shows the number of iterations needed to converge for all models of the test set. It does not increase with the size of the problem, measured as the number of binary variables. Combined with the fact that each iteration of IPE amounts to solve a linear program, this explains the good performance on the more difficult problems of the test set.

Algorithm	BBestim	BBbound	BBdepth	IPE	RR	SSR	CMSB	Relax&Fix
E	4.50	5.70	4.10	5.40	2.90	2.10	1.60	6.20
M	3.90	5.40	3.60	5.40	4.35	3.00	1.75	7.70
D	3.32	2.61	2.25	7.46	5.43	4.18	3.04	4.93

Table 3: Influence of problem difficulty on the ranking of the quality

6.1.2 IPE and cuts

Tables 4 and 5 show the influence of the cuts on the quality of the solution found by the different heuristics. Except for RR and SSR, the quality of the solution improves with the strength of the cuts. Notice the difference in the results in terms of partial average and rank for the XPRESS formulation between IPE and Relax&Fix. This comes from the fact that partial average does not take into account the XPRESS formulation of the SimEren problems for which Relax&Fix did not find a solution.

There is however a difference between the BBestim, BBbound, BBdepth and Relax&Fix on the one hand, and IPE, CMSB, SSR and RR on the other hand: the heuristics from the first group may be described as curtailed Branch&Bound schemes with specific node selection rules, while the other cannot. It is therefore hardly surprising that strong cuts improve the quality of the solution found by the procedure of the first group. What is more interesting is that other heuristics which do not rely on branching, as IPE and CMSB, show the same behaviour. The reason why this is not true for SSR and RR is that in these procedures, and more precisely in what makes them different from CMSB, the setup variables are fixed, and therefore the cuts have no influence anymore on the solution found.

Algorithm	BBestim	BBbound	BBdepth	IPE	RR	SSR	CMSB	Relax&Fix
LP	149.0%	124.2%	144.1%	111.1%	113.0%	126.4%	151.1%	116.7%
XPRESS	130.4%	115.0%	127.1%	110.4%	123.1%	127.1%	145.7%	106.4%
bc-prod	106.0%	103.0%	106.9%	104.4%	108.0%	112.2%	121.3%	101.1%

Table 4: Influence of the presence of cuts on the quality of the solution found: partial average

Algorithm	BBestim	BBbound	BBdepth	IPE	RR	SSR	CMSB	Relax&Fix
LP	2.73	2.91	2.18	7.05	5.59	4.05	2.86	6.27
XPRESS	3.68	4.18	2.77	6.82	4.50	3.68	2.50	5.09
bc-prod	5.36	5.86	4.79	4.71	3.29	2.00	1.29	7.43

Table 5: Influence of the presence of cuts on the quality of the solution found: rankings

6.1.3 On the usefullness of primal heuristics in a B&C framework

The discussion of the previous section shows clearly that heuristics as IPE, as opposed to branching rules, are especially important when the LP relaxation is weak, or the problem is large. Within the five minutes allowed, the best branching heuristic (BBestim) has only found a feasible solution of 79% of the problems, and 43% of the difficult ones. One reason is that when the formulation is weak, to fix a variable does not impose much constraint on the other variables. As a consequence, one has to branch many times to reach a integral feasible solution. But this makes more likely that no feasible or better solution will be found, because *one* bad branching choice may prevent finding good solutions. Indeed, this is a trivial consequence of the fact that the tree grows when the bounding device is weak.

This discussion indicates that branching is not the best technique to find good feasible solution for large problems for which we do not have tight formulations. This does not mean that branching should not be used anymore altogether even in that case, because, along with cutting, it is a powerful device to improve the dual bound. Rather, one should, *before selecting a node and branching*, decide wether it is more interesting to try to improve the primal or the dual bound. In the latter case, one can using cutting planes or branching on the node with the best dual bound. If the former is choosen, then a branching heuristic, IPE, Relax&Fix or any other problem-dependent heuristic scheme may be used. And the results presented in Section 6.1.1 show clearly that there are some cases when branching is not the best choice to improve the primal bound.

This is demonstrated on a fraction of the test set, namely the problems of Trigeiro (tr$x$-y) and the uncapacitated multi-level problems (multi$x$). Two formulations are considered for each problem: the standard formulation (1)-(6) and the one augmented with the cuts of XPRESS11. Two algorithms are compared, both running for a maximum time of 7200 seconds. The first one is BBbound, the default strategy which has been tested in Sections 6.1.1 and 6.1.2. The second is a best bound search, with the added feature that if the depth of the current node is a multiple of eight, BBIPE is performed. Note that both algorithms use a common mechanism to improve the dual bound, i.e. Branch&Bound with a best bound node selection rule, while they differ only in the way they generate feasible solutions: at each best bound node selection, BBbound dives in the branching tree until a better solution is found or two descendant nodes of the same parent are cut off, while BBIPE performs IPE if the depth of the node is a multiple of eight. The results can be found in Table 6, wherein the value of the solutions are expressed as a percentage of the best solution known.

Clearly, using IPE enables one to find better solutions faster. Indeed, the reason why the dual bounds are different, even though both procedures use the same mechanism to improve it, is that BBCEPH spent less time on searching for feasible solutions than BBbound.

	nb. of sol. found	first sol.	best sol.	best bound	gap
BBbound	3.32	120%	107%	62%	45%
BBIPE	3.74	106%	102%	63%	38%

Table 6: Improvement when using IPE in the tree for lotsizing problems tr$x$-y and multi$x$.

6.2 Network problems

Here, we would like to check if the same conclusions can be drawn for network design problems as for lotsizing ones. For each network design problem problem, we have run the default mp-opt with and without cuts, and IPE on the formulation (1)-(5), with and without the cuts generated by mp-opt with the default parameters. The branching strategy used is BBbound, which appeared to work best for lotsizing problems. It should be noted that Relax&Fix is no longer applicable, because there is no natural ordering the variables such as the time for production planning problems. Limited testing with a random ordering has proven unworkable due to a high number of problems with no solution at all. CMSB is still usable, but the two subsequent reduction subroutines cannot be adapted, again because of the lack of ordering of the variables. CMSB alone seemed as bad for network design problems as for lotsizing ones during early experiments, and no further testing seemed worthwhile.

The table 7 shows the results obtained, the value of the solution of mp-opt being the best solution found at the time IPE finished. We used the same measures of quality as for lotsizing problems, i.e. partial average and rankings. As opposed to lotsizing problems, both heuristics found (at least) one solution for nearly all problems. However, the feasibility problem of FCNF is easy: it suffices to check if the network flow problem with all arcs opened is feasible. Both BBbound and IPE ran into numerical difficulties for 3 problems. (see results in Appendix).

For models without cuts, IPE outperforms BBbound in terms of quality: it finds a better solution for 56% of the problems. Durations are difficult to compare, because for many problems, BBbound had not found a solution as good as IPE within the 300 seconds allowed.

When the XPRESS cuts are used, the two schemes appear to perform equally: IPE yields a better partial average, while BBbound outperforms IPE on 52% of the problems. As for lotsizing problems, one concludes that IPE performs comparatively well on big, loose formulations.

However, and in opposition to what was observed for lotsizing problems, adding cuts seems to make the finding of feasible solutions definitely more difficult. For both IPE and BBbound find worse solutions even though it took longer to find them (on average 4.15 longer) : on average, IPE yields solutions for models with cuts wich are 5.9% worse than models without, and 3.1% for BBbound.

Formulation	LP		LP + XPRESS cuts	
Algorithm	IPE	BBbound	IPE	BBbound
Partial average	107.0%	111.1%	112.6%	113.6%
Ranking	1.56	1.44	1.48	1.52

Table 7: Network design problems: a comparaison of IPE and BBbound.

7 Conclusion and future research

In this paper, we have presented a new heuristic for MIPs in which all integer variables are binary setup variables. This Converging Production Estimate Heuristic is particularly simple and can easily be implemented on various platforms while applicable to a broad class of problems.

We have compared IPE to other heuristics, including B&B search with various branching rules. The lotsizing test set includes multi-item multilevel capacitated lotsizing problems with setup times and families of products, and different formulations for each problems (with more or less strong cuts). IPE is the only heuristic which has found a feasible solution to all the problems within the five minutes allowed. IPE outperforms clearly other schemes for large problems with weak formulations, while for problems of small to medium size and/or good reformulations a Relax&Fix heuristic has proven better. Those two heuristics are also attractive because they do not require the tuning of parameters. The fact that the quality of the solution found by IPE improves with the strength of cuts generated makes it a potential device to generate good feasible solutions when doing Branch&Cut. This is has been shown experimentally. The experiments on network design problems confirm that IPE is a good choice for big problems, while showing that the use of XPRESS cuts slows down considerably the search for feasible solutions.

It remains to be studied if IPE could be adapted to other models. As mentioned in section 2, the generic model (1)-(5) does not contain such features as variable lower bounds on production, start-ups or changeovers. Production in batches of fixed size is also not part of it. More precisely, it would be interesting to know if IPE still converges quickly to a feasible solution in such cases. Other types of problems that can be modelled by (1)-(5) have not been tested, such as location problems. The same idea could be also be used for other purposes, for example the selection of the branching variable. Further testing of the approach is therefore needed.

References

[1] E. Arkin, D. Joneja, and R. Roundy. Computational complexity of uncapacitated multi-echelon production planning problems. *Operations Research Letters*, 8:61–66, 1989.

[2] G. Belvaux and L.A. Wolsey. Lotsizelib: A library of lot-sizing models and matrices. http://www.core.ucl.ac.be/wolsey/Lotsizel.htm.

[3] G. Belvaux and L.A. Wolsey. bc-prod: A specialized branch-and-cut system for lot-sizing problems. *Management Science*, 46(5):724–738, 2000.

[4] Gaëtan Belvaux. *Modelling and Solving Lot-Sizing Problems by Mixed Integer Programming.* PhD thesis, F.S.A. - Université catholique de Louvain, 1998.

[5] R. E. Bixby, S. Ceria, C. M. McZeal, and M. W. P. Savelsbergh. An updated mixed integer programming library: MIPLIB 3.0. *Optima*, 58:12–15, 1998. Problems available at *http://www.caam.rice.edu/bixby/miplib/miplib.html.*

[6] A.J. Clark and H. Scarf. Optimal policies for multi-echelon inventory problems. *Management Science*, 6:475–490, 1960.

[7] M. Constantino. *A Polyhedral Approach to Production Planning Models: Start-up Costs and Times, and Lower Bounds on Prodution.* PhD thesis, Faculté des Sciences, Université Catholique de Louvain, April 1995.

[8] M. Constantino. A cutting plane approach to capacitated lot-sizing with startup costs. *Mathematical Programming*, 75:353–376, 1996.

[9] M. Constantino. Lower bounds in lot-sizing models : a polyhedral study. *Mathematics of Operations Research*, 23(1):101–118, 1998.

[10] Cécile Cordier. *Development and experimentation with a Branch & Cut system for mixed integer programming.* PhD thesis, F.S.A. - Université catholique de Louvain, 1999.

[11] Dash Associates. *XPRESS-MP Optimisation Subroutine Library, Reference Manual, Release 11.* Blisworth House, Blisworth, Northants NN73BX, UK., 1999.

[12] Dash Associates. *XPRESS-MP, Reference Manual, Release 11.* Blisworth House, Blisworth, Northants NN73BX, UK., 1999.

[13] G.M. Guisewite and P. Pardalos. Minimum concave-cost network flow problems. *Annals of Operations Research*, 25:75–100, 1990.

[14] T. P. Harrison and H. S. Lewis. Lot sizing in serial assembly systems with multiple constrained resources. *Management Science*, 42(1):19–36, 1996.

[15] D. Hochbaum and A. Segev. Analysis of a flow problem with fixed charges. *Networks*, 19:291–312, 1996.

[16] E. Katok, H. S. Lewis, and T. P. Harrison. Lot sizing in general assembly systems with setup costs, setup times, and multiple constrained resources. *Management Science*, 44(6):859–877, 1998.

[17] D. Kim and P. M. Pardalos. A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. *Operations Research Letters*, 24:195–203, 1999.

[18] D. Klingman, A. Napier, and J. Stutz. Netgen: A program for generating large scale capacitated assignment, transportation, and minimum-cost flow network problems. *Management Science*, 20:814–820, 1974.

[19] T. Koch and A. Martin. Solving steiner tree problems in graphs to optimality. *Networks*, 32(3):207–232, 1998.

[20] R. Kuik, M. Salomon, and L. N. van Wassenhove. Batching decisions: Structure and models. *European Journal of Operational Research*, 75:243–263, 1994.

[21] J. Maes, J. O. McClain, and L. N. van Wassenhove. Multilevel capacitated lotsizing complexity and lp-based heuristics. *European Journal of Operational Research*, 53:131–148, 1991.

[22] K. Mehlhorn, S. Näher, M. Seel, and C. Uhrig. *The LEDA User Manual, Version 4.1.* `http://www.mpi-sb.mpg.de/LEDA/MANUAL/MANUAL.html`, 2000.

[23] H. Meyr. Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. *European Journal of Operational Research*, 120:311–326, 2000.

[24] A.J. Miller. *Polyhedral approaches to capacitated lot-sizing problems.* PhD thesis, Georgia Institute of Technology, 1999.

[25] A.J. Miller, G.L. Nemhauser, and M.W.P. Savelsbergh. Solving multi-item capacitated lot-sizing problems with setup times by branch-and-cut. CORE DP 2000/39, CORE, Université Catholique de Louvain, 2000.

[26] F. Ortega and L.A. Wolsey. A branch-and-cut algorithm for the single commodity uncapacitated fixed charge network flow problem. CORE DP 2000/49, CORE, Université Catholique de Louvain, 2000.

[27] Y. Pochet and L. A. Wolsey. Solving multi-item lot-sizing problems using strong cutting planes. *Management Science*, 37(1):53–67, 1991.

[28] Y. Pochet and L. A. Wolsey. Algorithms and reformulations for lot sizing problems. In W. Cook, L. Lovasz, and P. Seymour, editors, *DIMACS series in Discrete Mathematics and Computer Sciences*, pages 245–293. 1995.

[29] N. C. Simpson and S. S. Erenguc. Production planning in multiple stage manufacturing environments with joint costs, limited resources and set-up times. Technical report, Department of Management Science and Systems, University of Buffalo, 1998.

[30] H. Tempelmeier and M. Derstroff. A lagrangean-based heuristic for dynamic multilevel multi-item constrained lotsizing with setup times. *Management Science*, 42(5):738–757, 1996.

A Test instances

A.1 Lotsizing

NI and NT stand for the number of items and the number of time periods respectively. Then come the number of rows for the basic formulation, the number of rows for the XPRESS formulation and finally for the bc-prod formulation. A blank means inapplicable, either because of an echelon stock formulation or because the bc-prod cuts make the solution integer. #col and #ints are the total number of columns and the number of integer variables respectively. There are two entries in the #ints column for the SimEren problems because setup variables for individual items are better treated as continuous for BBs and Relax&Fix heuristic (leaving the family-setup variables the only binary variables, see [3]), while IPE and CMSB enforce them as integer. In columns 'cap', 'stp time' and 'mlt', a X indicates whether the problem is capacitated, include setup times and is multi-level respectively. Finally, the classification as Easy, Medium or Difficult of Section 5.1 is given.

Instance	NI	NT	#rows			#cols	#ints	cap	stp time	mlt	class		
			basic	xpress	bc-prod						basic	xpress	bc-prod
pp08a	8	8	137	228	252	240	64				M	E	E
set1ch	20	12	493	650	745	712	240	X			M	M	E
tr6-15	6	15	196	307	346	270	90	X	X		M	M	E
tr6-30	6	30	391	529	659	540	180	X	X		D	M	E
tr12-15	12	15	376	511	661	540	180	X	X		D	D	D
tr12-30	12	30	751	959	1361	1080	360	X	X		D	D	M
tr24-15	24	15	736	940	1339	1080	360	X	X		D	D	M
tr24-30	24	30	1471	1817	2779	2130	720	X	X		D	D	M
SimErenA1	72	16	3313	3491	4150	3648	192/960	X	X	X	M	M	M
SimErenA1en	72	16	4465	4680		3648	192/960	X	X	X	M	M	
SimErenA2	72	16	3313	3486	4181	3648	192 /960	X	X	X	M	M	M
SimErenA2en	72	16	4465	4679		3648	192/960	X	X	X	M	M	
SimErenB1	80	16	3905	4107	4803	3872	288/1024	X	X	X	D	D	M
SimErenB1en	80	16	4929	5145		3872	288/1024	X	X	X	D	D	
SimErenB2	80	16	3905	4092	4898	3872	288/1024	X	X	X	D	D	M
SimErenB2en	80	16	4929	5146		3872	288/1024	X	X	X	D	D	
multia	40	12	973	1209		1920	480			X	D	M	
multib	40	12	973	1190		1920	480			X	M	M	
multic	40	12	973	1201		1920	480			X	D	M	
multid	40	12	973	1734	1431	1920	480			X	M	M	M
multie	40	12	493	636	728	960	240			X	E	E	E
multif	40	12	373	494		720	180			X	E	E	

A.2 Network design

The type of the graphs used is given, along with the name, the number of nodes and arcs, the number of source nodes and demand nodes, the total amount of demand, and finally whether the variable costs are zero or not. The capacitated problems (g200x740, g55x188, k14x182, k16x240, p500x2988, p50x288, p30x160, r50x360) have the same structure as their uncapacitated counterpart and are not listed here.

Single-source problems

Type	Name	nodes	arcs	nsrc	ndem	totdem	c = 0
Steiner	beasleyC1	500	1250	1	4	4	X
	beasleyC2	500	1250	1	9	9	X
	beasleyC3	500	1250	1	82	82	X
	berlin	52	2652	1	15	15	X
	brasil	58	3306	1	24	24	X
	mc11	400	1520	1	212	212	X
	mc7	400	1520	1	169	169	X
	mc8	400	1520	1	187	187	X
Multi-Segment	beavma	89	195	70	1	14505	
	mtest4ma	100	975	70	1	3959	
	g150x1100	150	1100	1	50	1000	
	g150x1650	150	1650	1	50	1000	
	k15x420	15	420	1	14	95	
	k15x630	15	630	1	14	95	
	p50x576	50	576	1	30	968	
	p50x864	50	864	1	30	968	
MIPLIB	fixnet6	100	500	1	80	500	
Grid	g200x740c	200	740	1	30	10000	
	g200x740d	200	740	1	100	10000	
	g200x740e	200	740	1	150	10000	
	g200x740f	200	740	1	199	10000	
	g180x666	180	666	1	150	10000	
	g55x188c	55	188	1	30	1000	
$K_n + 1$	h50x2450	50	2450	1	49	277	
	h50x2450b	50	2450	1	49	303	
	h50x2450c	50	2450	1	44	252	
	h50x2450e	50	2450	1	44	253	
	h80x6320	80	6320	1	74	408	
	h80x6320b	80	6320	1	69	354	
	h80x6320c	80	6320	1	71	396	
	h80x6320d	80	6320	1	68	384	
Complete	k15x210	15	210	1	14	95	
	k20x380b	20	380	1	10	89	
	k20x380c	20	380	1	15	100	
	k20x380d	20	380	1	19	91	
	k20x380e	20	380	1	5	100	
Planar	p100x588c	100	588	1	5	673	
	p100x588d	100	588	1	5	673	X
	p200x1188c	200	1188	1	6	140	X
	p500x2988c	500	2988	1	6	606	
	p500x2988d	500	2988	1	6	606	X

Multi-source problems

Type	Name	nodes	arcs	nsrc	ndem	totdem	c=0
Grid	g200x740	200	740	100	10	1500	
	g200x740b	200	740	10	80	1000	
	g200x740g	200	740	100	100	1000	
	g200x740h	200	740	100	100	10000	
	g200x740i	200	740	100	100	200	
	g40x132	40	132	10	15	934	
	g50x170	50	170	13	18	1000	
	g55x188	55	188	17	14	1000	
Complete	k10x90	10	90	6	4	100	
	k14x182	14	182	4	6	1000	
	k14x182b	14	182	10	2	1000	
	k16x240	16	240	10	5	794	
	k16x240b	16	240	8	8	1000	
	k20x380	20	380	10	6	437	
Planar	p100x588	100	588	40	25	900	
	p100x588b	100	588	40	25	900	
	p200x1188	200	1188	40	25	900	
	p200x1188b	200	1188	40	25	900	
	p500x2988	500	2988	80	40	10000	
	p500x2988b	500	2988	80	40	10000	
	p50x288	50	288	10	15	500	
	p50x288b	50	288	10	15	500	
	p80x400	80	400	30	15	800	
	p80x400b	80	400	30	15	800	
Random	r20x100	20	100	8	8	1000	
	r20x200	20	200	8	8	1000	
	r30x160	30	160	10	8	1000	
	r50x360	50	360	30	15	100	
	r80x800	80	800	40	25	944	
Series-Parallel	sp100x200	100	200	14	11	51	
	sp150x300	150	300	11	13	90	
	sp150x300b	150	300	15	23	313	X
	sp150x300c	150	300	18	26	1174	
	sp150x300d	150	300	18	33	1135	X
	sp50x100	50	100	18	21	776	
	sp80x160	80	160	4	10	47	
	sp90x180	90	180	16	16	309	
	sp90x250	90	250	13	14	122	

B Complete computational results

The five following tables report the complete computational results.

Table 8 gives the value of the solutions found by the different heuristics, while the Table 9 gives the times for the same runs. Both tables are organized as follows: the first column is the name of the problem, the second specifies by which cut generator the basic formulation has been tightened, and the remaining gives the value of the solution found or the time in seconds needed by the considered heursitic. For the branching heuristics, the values of the solution found are the best solution found at the time IPE had finished, while the time is the time needed to find a solution at least as good as that of IPE.

In Tables 10, 11 and 12 are presented the complete computational results for the network design problems. The three Tables are organized as follows: the first column contains the name of the instance, the next group of four columns gives the value of the solution found by the routine specified by the first three rows, and the last four do the same for the times needed. As for lotsizing problems, the 'time' of BBbound is the time needed to find a solution at least as good as the one of IPE. There was no solution found by BBbound for g150x1100, g200x740d and g200x740e (XPRESS formulation) due to sensitivity to the numerical zero tolerances. Also, no solutions has been found by IPE for r80x800 and p500x2988b (XPRESS formulation) due to the difficulty of some LP relaxation. Those entries are indicated by "no sol.". All other blank entries mean that the initial LP relaxation was integer and there was thus no need for heuristics.

		BBestim	Bbbound	Bbdepth	IPE	RR	SSR	CMSB	Relax&Fix
pp08a	basic	8160	7720	8160	8040	8150	8320	9120	7580
	XPRESS	7710	7690	7710	8020	8620	8840.03	9030.01	7870
	bc-prod	7370	7370	7350	7790	8020	8120	8780	7460
set1ch	basic	60913	63236.5	60913	69945.75	63538.5	71295	88924.25	55977
	XPRESS	60761.8	58997.6	60761.8	60038	59157.3	61664.05	68801.56	55032
	bc-prod	54672.5	54672.5	54672.5	55420.5	58119.75	59900.25	61556	54672.5
tr6-15	basic	40641	no sol.	39762	40248	42263	43308	46183	40567
	XPRESS	39798.2	40852	40096.2	38976	40359.24	43787.21	44741.1	39777.25
	bc-prod	37801	38044	38502	39039	39154	40469	40469	37984
tr6-30	basic	no sol.	no sol.	no sol.	65632	69977	71594	78563	66059
	XPRESS	66754.2	65736	66754.2	63073	67939.37	69408.25	74126.13	63124.2
	bc-prod	63298	63369	63446	64370	65224	66068	69097	62459
tr12-15	basic	no sol.	no sol.	no sol.	79352.1	88607	97233	101741	83649
	XPRESS	no sol.	84186	no sol.	79825.3	90101.16	100721.1	102688.1	80730.1
	bc-prod	no sol.	78117	no sol.	78098	77615	77885	77885	74761.9
tr12-30	basic	no sol.	no sol.	no sol.	138650	no sol.	no sol.	no sol.	142488
	XPRESS	no sol.	144943	no sol.	136883.3	150214.3	151459.3	161850.3	142087
	bc-prod	132822	131384	132545	131988	136553	137290	142581	130957
tr24-15	basic	no sol.	no sol.	no sol.	144370	152342	157293	180928	146185
	XPRESS	no sol.	no sol.	no sol.	139593.6	160459.5	171061.4	174042.1	143541.1
	bc-prod	136968	136820	137268	137153	139122	140363	143012	136636
tr24-30	basic	no sol.	no sol.	no sol.	303800	324084	336098	382422	309084
	XPRESS	no sol.	no sol.	no sol.	301778.5	342587.9	360611.6	382258.4	no sol.
	bc-prod	288550	288097	288768	290006	293423	294349	297891	288035
A1C1S1	basic	13599.3	13642.8	13728.1	13458.1	13467.7	13884.7	17314.6	13559.5
	XPRESS	13249.3	13609.5	13737.2	12969.2	13652	13737.9	15984	no sol.
	bc-prod	12159.1	12055	12128.6	12440.2	12973.6	13339.6	14322.3	12029.16
A1C1S1en	basic	13597.9	13642.8	13726.6	13556.2	13408.3	13992.6	16965.6	13559
	XPRESS	13667.2	13929.1	13984.6	12722.4	14117	14303.2	16837.5	no sol.
A2C1S1	basic	13802.2	13124.3	13705.9	12984	13144.8	13740.5	17102.6	12216
	XPRESS	13564.4	12558.8	13302.6	13014.1	14514.8	14514.8	18346.4	no sol.
	bc-prod	11280	11310.3	11695.2	11467.7	11834.9	11838.4	13915	10963.9
A2C1S1en	basic	13973.4	13441.7	13705.9	13434.5	13882.3	14393.3	17055.3	12216
	XPRESS	12679.1	12779.9	13680.8	12862.7	14172.5	14172.5	17298.62	no sol.
B1C1S1	basic	46815	no sol.	no sol.	27089.1	27955.9	38119.6	61852.6	38800.41
	XPRESS	no sol.	no sol.	no sol.	29313.2	40787.1	40787.1	59436.2	no sol.
	bc-prod	25089.9	25396.9	28886	25497.2	29024.4	36779.1	48056	25395.72
B1C1S1en	basic	46381.3	no sol.	no sol.	29712.9	30596.2	48915.3	62957.2	38800.41
	XPRESS	39716.8	no sol.	no sol.	29769.7	45371.2	45371.2	50015.2	28502.42
B2C1S1	basic	34327.7	no sol.	no sol.	29225.6	30089.9	43707.2	59717	33591.7
	XPRESS	no sol.	no sol.	no sol.	32003.9	45348.9	45922.8	53295.5	32160.55
	bc-prod	30964.5	30272.8	28689.8	31037.5	30868.3	33258.5	38994.6	27973.3
B2C1S1en	basic	38052.1	no sol.	no sol.	32774.9	33727.1	48568.9	59457.1	33540.22
	XPRESS	42937.3	no sol.	no sol.	31668.5	no sol.	no sol.	no sol.	no sol.
multia	basic	8353	8454	8353	5219	4702	4702	5899	6597
	XPRESS	6277	6641	6277	5034	5050.03	5050.03	7862	4970
multib	basic	6004	5887	6004	4318	4203	4203	4318	4376
	XPRESS	4591	5013	4591	4447	3911.01	3911.01	4375.01	3788
multic	basic	6686	5934	6686	4255	3982	3982	4819	5066
	XPRESS	5281	5578	5281	4646	4330.03	4330.03	6244	4000
multid	basic	33830.8	13523	33830.8	12802.5	13037.76	13037.76	13420.25	13446
	XPRESS	13686.8	12876	13686.8	13463	12998.2	12998.2	13034.57	12907.6
	bc-prod	18283.8	13438.1	18283.8	13348.5	202690	214085	214085	12875
multie	basic	49754.25	2625.5	49754.25	2625.5	2797.75	2797.75	2797.75	2625.5
	XPRESS	2701.36	2673	2701.4	2673	3407.09	4219.58	4219.58	2949.2
	bc-prod	2761.75	2761.75	2761.75	2749.75	2992	2992	2992	2633.5
multif	basic	89076	1444	89076	1440	1473	1473	1484	1440
	XPRESS	3818.8	1473	3818.8	1440	1473	1473	1473	1440

Table 8: Lotsizing problems: value of the solutions

		Bbestim	Bbbound	Bbdepth	IPE	RR	SSR	CMSB	Relax&Fix
pp08a	basic	11	0.47	0.3	0.49	1.3	0.49	0.14	0.5
	XPRESS	0.5	0.4	0.5	0.48	1.47	0.73	0.32	8.6
	bc-prod	0.3	0.5	0.6	0.74	1.57	0.91	0.47	0.9
set1ch	basic	1	2.5	1	0.84	9.09	2.56	0.28	0.91
	XPRESS	4.5	1.3	1	1.56	9.76	2.81	0.44	3.05
	bc-prod	0.5	0.7	1.1	1.26	8.38	2.9	0.94	1.08
tr6-15	basic	0.7	8.5	1.2	1.06	2.13	1	0.17	0.66
	XPRESS	4.3	123	1	1.2	3.36	2.05	0.32	2.36
	bc-prod	1	0.85	1	1.31	3.19	1.24	0.32	1.27
tr6-30	basic	no sol.	1.5	no sol.	2.54	5.12	2.55	0.28	1.81
	XPRESS	no sol.	1	1	2.76	7.24	2.99	0.31	4.54
	bc-prod	1.1	1.8	1.5	1.7	9.56	3.6	0.59	3.92
tr12-15	basic	no sol.	no sol.	no sol.	2.49	7.32	3.66	0.22	1.46
	XPRESS	no sol.	181	no sol.	3.44	10.43	4.73	0.44	4.2
	bc-prod	no sol.	5.5	no sol.	3.53	11.43	5.56	0.99	2.94
tr12-30	basic	no sol.	no sol.	no sol.	6.58	no sol.	no sol.	no sol.	4.09
	XPRESS	no sol.	216	no sol.	9.12	14.47	9.13	1.33	12.5
	bc-prod	6.5	2.5	3.5	6.03	33.83	14.02	2.72	13.12
tr24-15	basic	no sol.	no sol.	no sol.	8.34	21.65	9.88	0.65	1.91
	XPRESS	no sol.	no sol.	no sol.	8.82	25.25	11.31	0.98	21.6
	bc-prod	5.2	3	5.8	6.01	33.08	14.12	1.87	5.32
tr24-30	basic	no sol.	no sol.	no sol.	32.73	73.19	32.49	1.13	7.69
	XPRESS	no sol.	no sol.	no sol.	35.65	85.06	37.87	2.19	no sol.
	bc-prod	6.5	6.5	14	21.15	129.91	63.07	6.18	18.9
A1C1S1	basic	144	57	17	28.8	499	313.6	17.64	34.5
	XPRESS	279	263	13	52.2	648.1	419.3	11.81	no sol.
	bc-prod	38	39	65	201.5	1864.5	1141.5	56.8	197.73
A1C1S1en	basic	35	64	20	31.5	496.5	360.1	19.92	38.8
	XPRESS	295	176	48	54	688.8	462.5	21.5	no sol.
A2C1S1	basic	46	159	12	29.2	428	276.5	13.71	45.9
	XPRESS	32	16	9	40.2	456.2	289.4	14.46	no sol.
	bc-prod	87	106	189	190	1545.1	837.1	48.62	146.95
A2C1S1en	basic	258	25	12	23.1	388	286.9	13.72	46.1
	XPRESS	37	18	13	49.8	851.9	693.9	17.88	no sol.
B1C1S1	basic	296	no sol.	no sol.	22.9	713.8	418.2	17.36	42.68
	XPRESS	no sol.	no sol.	no sol.	65.6	1099.7	569.5	30.48	no sol.
	bc-prod	113	176	50	224.17	3377.8	1596.3	100.5	159
B1C1S1en	basic	275	no sol.	no sol.	29.6	762.5	477.6	21.55	46.21
	XPRESS	93	no sol.	no sol.	49.6	1052.4	575.4	20.86	49.9
B2C1S1	basic	93	no sol.	no sol.	55	982.8	563.4	26.87	53.8
	XPRESS	no sol.	no sol.	no sol.	62.5	1565.5	916.2	35.11	76.31
	bc-prod	86	78	97	296.9	3923.1	1984.7	106.88	228.4
B2C1S1en	basic	49	no sol.	no sol.	32.9	1080.2	635.8	30.94	62.65
	XPRESS	220	no sol.	no sol.	58.4	no sol.	no sol.	no sol.	no sol.
multia	basic	70	11	5	0.64	21.49	5.23	0.35	2.31
	XPRESS	64	83	3	0.83	39.49	5.71	0.69	8.33
multib	basic	17	155	6	0.47	27.45	6.97	0.5	2.31
	XPRESS	12	8	4	0.91	37.9	7.79	0.55	17.9
multic	basic	26	11	5	1.46	28.66	5.01	0.57	2.44
	XPRESS	9.5	38	4	0.93	40.42	6.18	0.79	23.36
multid	basic	16	28	8	0.85	73.92	24.61	0.49	9.41
	XPRESS	47	12	10	2.02	102.77	30.86	0.96	24.99
	bc-prod	54	16	4	5.08	98.35	18.36	4.59	14.3
multie	basic	10	3	2	0.35	12.9	5.82	0.18	2.45
	XPRESS	5	1.7	2	0.67	25.96	7.47	0.54	4.65
	bc-prod	2.1	2.5	1	1.6	23.96	6.41	0.75	4.97
multif	basic	7	2	1	0.3	6.76	2.66	0.13	1.62
	XPRESS	2.2	1	1	0.55	8.93	3.51	0.14	0.86

Table 9: Lotsizing problems: times

	Value				Times			
	basic		XPRESS		basic		XPRESS	
	BBbound	IPE	BBbound	IPE	BBbound	IPE	BBbound	IPE
beasleyC1	92	102	87	124	0.5	4.66	39	11.91
beasleyC2	167	171	167	193	1	8.11	1	9.92
beasleyC3	825	814	864	887	160	7.34	5	11.83
berlin	1542	1300	1906	1595	>300	50.53	>300	115.68
brasil	20047	16569	25619	27352	>300	281.25	51	115.57
mc11	12351	13057	23009	12979	6	8.89	>300	20.9
mc7	3989	4353	4428	4267	6	11.16	63	19.01
mc8	1656	1717	3170	1815	7	10.95	>300	24.64
beavma	410860	399425	383746	383285	161	0.46	0.5	0.37
mtest4ma	60342	52768	53679	53127	>300	2.04	22	4.18
g150x1100	84407	79492	no sol.	88826	>300	5.72	no sol.	141.05
g150x1650	85250	74691	76023	82778	>300	13.64	2	244.71
k15x420	875	843	819	909	35	1.51	0.5	9.79
k15x630	993	1037	936	947	1	2.34	0.5	2.93
p50x576	20462	19727	19742	19627	>300	1.86	3	2.93
p50x864	20236	19776	19173	19007	51	2.41	1	0.81
fixnet6	11012	4296	6316	4284	>300	1.84	49	3.22
g180x666	629603	638930	639783	633839	1	1.4	>300	4.47
g200x740c	681920	681972	681079	680624	0.7	0.73	18	1.34
g200x740d	589241	589150	no sol.	586227	5	1.36	no sol.	2.09
g200x740e	604423	604179	no sol.	600714	>300	1.59	no sol.	4.24
g55x188c	39085	36690	35464	35509	>300	0.53	0.5	0.51
h50x2450	437328	458026	553144	489175	1	24.61	>300	44.04
h50x2450b	56.09	53.8	68.22	67.27	>300	21.2	153	44.96
h50x2450c	3168	3184	3621	3497	1	53.24	>300	129.06
h50x2450e	3190	3194	3680	3843	36	51.48	1	128.74
h80x6320	5087	4913	6134	5541	>2000	414.13	>3000	1787
h80x6320b	4502	4293	4882	4811	>2000	426.59	>3000	1291
h80x6320c	4755	4704	5281	5056	>2000	409.65	>3000	1250.1
h80x6320d	5118	4796	5698	5433	>2000	408.03	>3000	1454.3
k15x210	17820	18244	16128	16180	0.5	2.01	0.5	0.93
k20x380b	11949	12337			1	3.37		
k20x380c	19374	18791	17159	22385	18	3.96	1	26.94
k20x380d	22546	25977	20979	20979	0.5	3.71	1	1.39
k20x380e	7377	7818			1	2.03		
p100x588c	173598	180220	172770	173742	3	6.43	0.5	7.86
p100x588d	5	5	6	6	0.5	4.9	4	6.03
p200x1188c	15531	15747	15078	29037	0.5	12.42	0.5	48.02
p500x2988c	15215	15215	15323	15215	9	9.06	9	1.57
p500x2988d	6	6	6	10	2	53.74	0.5	219.8

Table 10: Uncapacitated single-source network design problems: results

	Value				Times			
	basic		XPRESS		basic		XPRESS	
	BBbound	IPE	BBbound	IPE	BBbound	IPE	BBbound	IPE
g200x740	45301	46066	45022	44734	1	3.77	136	14.09
g200x740b	183819	183253	182513	181256	>300	1.9	>300	8.11
g200x740g	57121	49093	59279	48065	>300	4.62	>300	107.65
g200x740h	137235	136966	int infeasible	134583	7	5.58		53.79
g200x740i	43439	34002	44080	34343	>300	4.05	>300	100.27
g40x132	27818	27588	27432	27484	3	0.8	1	1.89
g50x170	31631	28765	30209	26072	20	0.94	52	0.88
g55x188	29296	26808	27966	25327	>300	0.87	46	1.83
k10x90	576	589	579	582	0.1	0.97	0.5	0.61
k14x182	8615	8491	8570	10465	37	3.83	0.3	10.5
k14x182b	11260	11561	11064	12379	1	1.72	0.3	2.12
k16x240	11028	11715	11655	14630	0.2	5.55	0.5	26.86
k16x240b	12444	13552	13139	13564	0.2	5.46	4	27.73
k20x380	2038	2177	2025	2695	0.1	4.95	0.5	36.55
p200x1188	11716	12313	12023	12297	1	16.68	9	156.77
p200x1188b	70416	65363	65382	65777	>300	15.67	47	233.65
p500x2988	72267	72287	73200	72662	3	9.09	>300	263.24
p500x2988b	192524	197333		no sol.	4	89.12		no sol.
p50x288	6377	6522	6216	6447	0.1	2.4	1	5.2
p50x288b	25389	23261	22440	23133	>300	2.34	0.5	5.67
p80x400	8796	8855	8939	8748	0.2	2.71	>300	10.04
p80x400b	45523	43387	42850	43685	>300	3.66	14	15.33
r20x100	17255	17335	16528	17205	1	1.51	1	1.52
r20x200	15151	16934	17626	17523	0.2	4.38	13	10.95
r30x160	24008	24244	24078	24127	1	1.69	1	2.31
r50x360	1885	1803	1836	1914	>300	3.42	13	72.07
r80x800	5447	5645		no sol.	0.5	14.39		no sol.
sp100x200	38503	35209	37356	34805	>300	0.6	3	0.38
sp150x300	32289	30918	34766	33178	>300	0.75	6	1.35
sp150x300b	58	60	58	60	0.5	1.44	0.5	2.56
sp150x300c	585371	590384	579970	572297	0.5	0.6	2	0.66
sp150x300d	69	71	72	71	0.5	0.98	4	1.6
sp50x100	51129	51489	50968	50968	0.3	0.33		
sp80x160	20245	22124	19549	19549	0.3	0.42	0.21	0.21
sp90x180	70588	69236	69798	68862	>300	0.86	1	0.33
sp90x250	28867	23844	23571	23571	>300	0.72	0.23	0.23

Table 11: Uncapacitated multi-source network design problems: results

	Value				Times			
	basic		XPRESS		basic		XPRESS	
	BBbound	IPE	BBbound	IPE	BBbound	IPE	BBbound	IPE
g200x740j	50637	48256	47613	47944	>300	2.98	14	22.96
g55x188d	35551	30699	29407	30215	134	0.56	1	1.42
k14x182c	21379	21103	19590	20718	3	1.76	1	3.72
k16x240c	17727	16284	14794	18496	91	1.41	3	9.74
p500x2988e	1E+09	72016	74261	72152	>300	6.66	>300	27.65
p50x288c	10497	10356	9844	10040	3	1.31	1	2.42
p80x400b	27928	23618	23519	23311	>300	2.07	5	2.41
r50x360b	2267	1879	1852	1872	>300	1.65	6	11.34

Table 12: Capacitated network design problems: results