An Integer Local Search Method with Application to Capacitated Production Planning

Joachim P. Walser[1]
Programming Systems Lab
Univ. des Saarlandes, Postfach 151150
66041 Saarbrücken, Germany
walser@ps.uni-sb.de

Ramesh Iyer
i2 Technologies
909 East Las Colinas Boulevard
Irving, TX 75039
Ramesh_Iyer@i2.com

Narayan Venkatasubramanyan
i2 Technologies
909 East Las Colinas Boulevard
Irving, TX 75039
narayan@i2.com

Abstract

Production planning is an important task in manufacturing systems. We consider a real-world capacitated lot-sizing problem (CLSP) from the process industry. Because the problem requires discrete lot-sizes, domain-specific methods from the literature are not directly applicable. We therefore approach the problem with WSAT (OIP), a new domain-independent heuristic for integer optimization which generalizes the Walksat algorithm. WSAT (OIP) performs stochastic tabu search and operates on over-constrained integer programs. We empirically compare WSAT (OIP) to a state-of-the-art mixed integer programming branch-and-bound solver (CPLEX 4.0) on real problem data. We find that integer local search is considerably more robust than MIP branch-and-bound in finding feasible solutions in limited time, and branch-and-bound can only solve a sub-class of the CLSP with discrete lot-sizes. With respect to production cost, both methods find solutions of similar quality.

Introduction

Production planning is an important task in manufacturing systems and gives rise to a variety of optimization problems. Here we study a real-world lot-sizing problem from the process industry (manufacturing of chemicals, food, plastics, etc.). The problem is expressed as follows: given a set of products and a collection of customer orders with due dates, construct a minimal-cost production plan such that all orders are met in time without exceeding resource capacity. The total cost of a plan consists of inventory and labor costs.

The problem under consideration is similar to the well-studied capacitated lot-sizing problem (CLSP, see (Drexl & Kimms 1997) for a survey) but includes the requirement of discrete lot-sizes that prevents a direct application of domain-specific methods from the literature (Diaby *et al.* 1992; Kirca & Kökten 1994; Hindi 1996). We therefore approach the problem with a new domain-independent heuristic for integer optimization, WSAT (OIP), and empirically compare it to a commercial mixed integer programming (MIP) branch-and-bound solver (CPLEX 4.0).

The first part of the paper introduces the WSAT (OIP) heuristic and the constraint class on which it operates, over-constrained integer programs (OIPs). WSAT (OIP) is a straightforward extension of the WSAT $(\mathcal{PB})$ heuristic (Walser 1997) from binary variables to variables ranging over finite integer domains. While both methods generalize the stochastic Walksat algorithm for propositional satisfiability (Selman, Kautz, & Cohen 1994), their refined strategy for move selection follows principles from tabu search (Glover & Laguna 1993). By using an algebraic problem specification as input, such *integer local search* methods are potentially applicable to a range of optimization problems of practical importance. This calls for investigating the effectiveness of such heuristics by empirical comparison with established methods.

The second part of the paper describes a case study of WSAT (OIP) on a large CLSP with discrete lot-sizes and fixed charges. We compare the experimental results on real data to CPLEX applied to a tight integer programming model. We find that MIP branch-and-bound can only solve a sub-class of the CLSP with discrete lot-sizes, namely the problem where fixed charges and lot-sizes are equal. Further, WSAT (OIP) is considerably more robust than CPLEX in finding feasible solutions in limited time, in particular as the capacity constraints are tightened. With respect to production cost, both methods find solutions of similar quality. We examine fixed-capacity and varied-capacity problems. Using a Lagrangean relaxation technique we provide lower bounds that prove that the fixed-capacity problems are solved with near-optimal overall cost. We show that substantial savings can be achieved by varying capacity.

Part I. Integer Local Search

Many domain-specific heuristics for problem classes like set-covering, generalized assignment, or time-tabling exist in the operations research literature. In contrast, only few general purpose heuristics for integer programming

[1]This study was carried out during a visit of the first author at i2 Technologies.

have been described that aim at covering a broader range of combinatorial problems. These heuristics are of two kinds, (i) approaches based on linear programming which relax the integrality constraints (Aboudi & Jörnsten 1994; Løkketangen, Jörnsten, & Storøy 1994; Glover & Laguna 1997; Balas & Martin 1980), and (ii) techniques in which local moves are performed directly in the space of integer solutions, such as simulated annealing (Connolly 1992; Abramson, Dang, & Krishnamoorthy 1996) and stochastic local search (Walser 1997). We will refer to the second class of heuristics as *integer local search* methods.

In this paper we generalize WSAT $(\mathcal{PB})$ (Walser 1997) from 0-1 variables to finite domain integer variables, introducing WSAT (OIP). For reasons of its variable selection strategy, WSAT (OIP) operates on over-constrained integer programs (OIPs) instead of classical integer programs (Nemhauser & Wolsey 1988). Unlike integer programs (IPs), OIPs represent the overall optimization objective by competing sub-objectives instead of using a single objective function. Similarly as discussed in the context of constraint hierarchies (Borning, Freeman-Benson, & Wilson 1996), OIPs encode optimization objectives with soft constraints.

Over-Constrained Integer Programs

We define a constraint system of hard and soft inequalities and equations over integer variables as an *over-constrained integer program*. Here, we consider the special case where all constraints are linear and the system can be denoted in matrix notation as

$$\begin{aligned} A\mathbf{x} &\geq \mathbf{b} \\ C\mathbf{x} &\leq \mathbf{d} \;\; (\textit{soft}) \\ x_i &\in D_i. \end{aligned} \tag{1}$$

A and C are real valued coefficient matrices, $\mathbf{b}, \mathbf{d}$ are real-valued vectors, and $\mathbf{x}$ is the variable vector, all variables x_i ranging over finite integer domains D_i.[1] The objective is to minimize some measure of the overall violation of soft constraints, subject to the hard constraints. Given an evaluation function $\|.\|$ to measure the overall violation of soft constraints, (1) is interpreted as the following optimization problem.

$$\begin{aligned} \min \;\; & \|C\mathbf{x} - \mathbf{d}\| \\ \text{subject to} \;\; & A\mathbf{x} \geq \mathbf{b}, \;\; x_i \in D_i. \end{aligned} \tag{2}$$

The following sections describe WSAT (OIP), a local search heuristic to find approximately optimal solutions to over-constrained IPs.

[1] We will refer to a problem of the form (1) as being in *min normal form*. Every OIP minimization problem can be converted into min normal form by multiplying every 'incorrect' inequality (e. g. $\leq$ instead of $\geq$) by -1 and converting every equality into two inequalities. Input to WSAT (OIP) is not required to be in min-normal form.

The Variable Selection Cycle of WSAT (OIP)

WSAT (OIP) is an iterative repair heuristic. Starting with a random variable assignment, individual variable/value pairs are iteratively selected to be changed, thereby moving in the space of feasible and infeasible solutions. Generalizing from the Walksat algorithm (Selman, Kautz, & Cohen 1994), variable changes are selected in a two-stage process of first randomly selecting an unsatisfied (hard or soft) constraint for partial repair and within the constraint selecting a variable to be changed. The criterion for move selection is to perform hill-climbing on a *score* which reflects both the degree of infeasibility and the optimization objective.

A move of WSAT (OIP) consists of *triggering* the value of a finite domain integer variable to a smaller or greater value close to its current assignment. This extends WSAT $(\mathcal{PB})$ in which Boolean variables are *flipped* (complemented). Occasionally, a restart with a new initial assignment takes place to escape from local optima, typically after a fixed number of moves. To describe the move selection strategy for over-constrained IPs in more detail, we first need a score definition. Given a particular assignment $\mathbf{x}$, we define a score to evaluate a system of the form (1) as

$$\textit{score}(\mathbf{x}) = \|\mathbf{b} - A\mathbf{x}\|_\lambda + \|C\mathbf{x} - \mathbf{d}\| \tag{3}$$

We employ a simple evaluation function $\|.\|$ which scores each violated constraint in proportion to its degree of violation: $\|\mathbf{v}\| := \sum_i \max(0, v_i)$. Additionally, the score computation (3) uses a vector $\lambda \geq 0$ for weighting the violations of hard constraints, defined by $\|\mathbf{v}\|_\lambda := \sum_i \max(0, \lambda_i v_i)$. These weights can be statically assigned or dynamically updated during the search (Selman & Kautz 1993). The reported experiments were all performed with statically assigned weights.

Observation of the two-stage move selection strategy motivates the use of OIP encodings for integer local search: In the constraint/variable selection, the selected constraint induces a choice of moves leading towards a local goal (i. e. satisfying the constraint). In contrast with standard IP encodings, decision alternatives in OIPs are grouped together within one sub-objective and are evaluated in direct competition. We hypothesize that this helps to focus the search.

Local Moves

The remaining degrees of freedom are how to select a variable from within a clause and which new value to assign to it. The fundamental principle behind WSAT (OIP) is greediness: Select local moves that most improve the total score. Additionally, adaptive memory (Glover & Laguna 1993) and noise are employed to overcome local minima. Figure 1 outlines the variable selection strategy in detail. As has been reported for SAT local search (McAllester, Selman, & Kautz 1997; Parkes & Walser 1996), the details of the variable selection are important for performance. The described strategy includes a tabu mechanism (Glover

1. Randomly select an unsatisfied constraint α (with probability p_{hard} a hard constraint, and with $1 - p_{\text{hard}}$ a soft constraint).
2. From α, select all variables which can be changed such that α's score improves. For each such variable, select one or more α-improving values and compute the hypothetical total scores (finite domain integer variables are triggered up or down, Boolean variables are flipped).
3. From the selected variable-value pairs, remove the ones which are *tabu* (tabu-aspiration by score).
4. Of the remaining variable-value pairs, select one which most improves the total score, if assigned. Break ties according to i) *frequency* and ii) *recency*
5. Only if the total score cannot be improved: With probability p_{noise}, select a random α-improving non-tabu variable-value pair. With $1 - p_{\text{noise}}$, select the best possible one.

Figure 1: A move selection strategy for WSAT (OIP).

& Laguna 1993) with tenure of size t: No variable-value pair may be assigned that has been assigned in the previous t moves. Further, all ties between otherwise equivalent variable-value pairs are broken by a history mechanism: On ties, choose the move that was chosen i) least frequently, and then ii) longest ago. The experimental results section reports on parameter settings.

Part II. Production Planning

The problem under consideration can be classified as single-level, dynamic-demand capacitated lot-sizing problem (CLSP) with discrete lot-sizes and fixed charges. Given is a set of products and a number of customer orders (or forecasted demands) with due dates on a finite planning horizon. The goal is to compute a minimal-cost production plan such that all customer orders are met in time. No lateness or shortage of orders is permitted. Products (or *items*) can be produced in discrete periods of the planning horizon (weeks). Because production consumes resources and resources have limited capacity, items often have to be produced earlier than needed and carried to the period where they are shipped. Such carrying incurs inventory cost (opportunity cost of capital and storage cost) which is one of two cost factors in the problem considered here. Solving the CLSP optimally is known to be NP-hard (Bitran & Yanasse 1982). Table 1 specifies the problem parameters.

The CLSP considered here has two particularities: (i) Items can only be produced in predefined quantities (lots) and setup costs are compensated by economic production quantities (EPQs). At any time, production of item i is possible in quantities of 0 or $E_i + k \cdot L_i$, where $k \geq 0$, L_i is the

Index	Definition
i	Index for items/products.
t	Index for time periods.
Symbol	**Definition**
L_i	Lot-size of product i.
E_i	Economic production quantity of product i.
D_{it}	Demand of product i in time period t.
T_t	Total labor units available in time period t.
R_i	Unit labor requirement for product i.
C_i	Cost of carrying product i per unit/period.
Ω_{it}	Future demand of product i starting period t.
T	Number of periods.
N	Number of items.
S	cost per labor shift.

Table 1: Parameters for the CLSP with discrete lot-sizes and fixed charges (EPQs).

lot-size and E_i is the EPQ for item i (every EPQ is a multiple of the lot-size). (ii) The only resource is labor, available in either one or two shifts in any period. The amount of available labor has an associated cost (labor availability and consumption are expressed in cost units). Thus, production cost is equal to the sum of labor and inventory costs.

In the problem, labor capacity can be varied between one and two shifts. Because less capacity enforces earlier production of items, a tradeoff exists between labor and inventory costs. Because labor costs dominate inventory costs, reducing labor is critical to substantially save costs. However, due to practical considerations it is not acceptable to have too many labor level changes; thus the number of labor level changes considered was limited to 2 in our experiments. To optimize the overall problem, we take the approach to solve a series of capacitated lot-sizing problems with different 'labor profiles' and choose the best solution, as follows.

Labor Profiles Labor consumption varies between items and is expressed by parameters R_i in terms of resource consumption per production of one unit of item i. In any period t, the total labor consumption is limited by T_t, available in one or two shifts. One shift incurs a per-week cost of S, two shifts incur $2S$. A labor profile thus corresponds to a set $\{(t, T_t) \mid 1 \leq t \leq T, T_t \in \{S, 2S\}\}$. Possible labor profiles are restricted to the pattern 2-shifts/1-shift/2-shifts and can be denoted by an interval $[s_1, s_2]$ referring to periods $s_1 \ldots s_2$ on one shift, and periods $1 \ldots s_1 - 1$ and $s_2 + 1 \ldots T$ on two shifts. The cost of a labor profile $[s1, s2]$ is thus $(T - (s2 - s1 + 1)) \cdot 2S + (s2 - s1 + 1) \cdot S$.

Every labor profile has an optimal inventory cost. If labor could be freely varied, the labor availability would have to be modeled with problem variables. However, since the number of allowed labor profiles is small, we factored the labor variability out from the optimization problem and approached the problem by solving each permitted labor profile, optimizing one CLSP at a time. Possible shift boundaries $[s_1, s_2]$ were generated starting with $s_1 = 1$ and an initial one-shift period length l ($s_2 = s_1 + l - 1$). Iteratively, s_2 was then increased as long as WSAT (OIP) found feasible solutions for the resulting CLSP (for CPLEX, as long as infeasibility was not proved). If no feasible solution was found (for CPLEX, if infeasibility of the profile was proved), s_1 was increased to the next period and s_2 was reset.

The two different integer solvers require different algebraic models which are described in the following.

Integer Local Search Model

The integer local search model is straightforward. Production quantities per item and time period are expressed by finite domain variables p_{it} that range over the allowed production quantities (and are bounded by the summed future demand Ω_{it}):

$$p_{it} \in \{p \leq \Omega_{it} \ | \ p = 0 \ \vee \ p = E_i + k \cdot L\}$$

where $k = 0, 1, 2, \ldots$, for every item i and time period t and Ω_{it} is determined as $\Omega_{it} = \sum_{t \leq s \leq T} D_{is}$.

To formulate the constraints, we will make use of the abbreviation $S[i,t]$ representing the amount of product i carried in inventory in time period t (textually substituted in the constraints):

$$S[i,t] = \sum_{s=1}^{t} p_{is} - D_{is}$$

The formulation is as follows.

$$S[i,t] \geq 0 \quad \forall i, t \qquad \text{(NOH)}$$

$$\sum_i R_i \cdot p_{it} \leq T_t \quad \forall t \qquad \text{(CAP)}$$

$$\text{soft: } C_i \cdot S[i,t] \ \leq \ 0 \quad \forall i, t \qquad \text{(INV)}$$

Negative-on-hand constraints (NOH) ensure that all orders are met in time. Capacity constraints (CAP) express that available labor capacity is not to be exceeded. The soft constraints (INV) express the competing objectives of minimizing inventory costs; for every item and time period, the inventory cost from carrying material has to be minimized. For every feasible solution, the resulting objective (the total inventory cost) is the summed violation of all soft constraints measured by our definition of $\|.\|$ in (2). Using finite domain variables to model production, the local search progresses by moving production up or down in allowed quantities induced by the violated constraints.

0-1 Integer Model The first modeling attempt used an over-constrained 0-1 integer model with a logarithmic encoding of production quantities ($E_i x_1 + L_i x_2 + 2L_i x_3 + 4L_i x_4 + \ldots$). In addition to the blowup of the number of variables for this model, running WSAT ($\mathcal{PB}$) did not yield solutions of acceptable quality. We put this failure down to the fact that with a logarithmic encoding, a small change of production often requires a long sequence of local moves. For example, an increase from $2^k - 1$ to 2^k lots can only be achieved by flipping $k + 1$ variables. This appeared to be a strong hindrance of the search process.

	Sets
SKU	Set of products (stock keeping units).
SKU_1	Set of products for which lot-size (L_i) is equal to economic production quantity (E_i).
SKU_2	Set of products for which lot-size is a multiple of economic production quantity.
	Variables
s_{it}	Amount of product i carried in inventory in time period t.
x_{it}	Amount of product i produced in time period t.
y_{it}	Number of lots of product i produced in time period t.
z_{it}	Binary variable which is unity if product i is produced in time period t.

Table 2: Sets and decision variables for the MILP model.

Mixed Integer Programming Model

This section requires some familiarity with integer programming terminology, as covered for example in (Nemhauser & Wolsey 1988). The sets and variables defined in the mixed integer programming model (MILP) are given in tables 1 and 2. The problem formulation (P) is as follows.

$$\mathbf{P}: \min_{x_{it}, y_{it}, z_{it}, s_{it}} \sum_{i=1}^{N} \sum_{t=1}^{T} C_i s_{it} \qquad (4)$$

subject to

$$x_{it} + s_{i,t-1} = D_{it} + s_{it} \quad \forall i, t \qquad (5)$$

$$x_{it} = L_i y_{it} \quad \forall i \in SKU_1 \qquad (6)$$

$$x_{it} = E_i z_{it} + L_i y_{it} \quad \forall i \in SKU_2 \qquad (7)$$

$$E_i z_{it} \leq x_{it} \leq \Omega_{it} z_{it} \quad \forall i \in SKU_2 \qquad (8)$$

$$\sum_{k=1}^{t} x_{ik} \geq L_i \lceil \sum_{k=1}^{t} D_{ik} / L_i \rceil \quad \forall i \in SKU_1, t \qquad (9)$$

$$\sum_{k=1}^{t-1} x_{ik} \geq \sum_{k=1}^{t-1} D_{ik} z_{it} + \sum_{k=1}^{t} D_{ik}(1 - z_{it}) \quad \forall i \in SKU_2, t \qquad (10)$$

$$\sum_i R_i x_{it} \leq T_t \quad \forall t \qquad (11)$$

$$z_{it} \in \{0, 1\}, y_{it} \text{ integer}$$

In the MILP model, equation (4) represents the sum of total inventory carrying costs. Equation (5) is the material balance in each time period and equations (6)-(7) determine the total production quantity of each product in time period t. Note that binary variables are only defined for $i \in SKU_2$. Equation (8) states that if z_{it} is non-zero, then the minimum amount (EPQ) must be produced, and cannot exceed the bound Ω_{it} (only for items in SKU_2).

Equations (9)-(10) represent constraints that tighten the relaxation gap between the integer solution and the LP relaxation of the problem. Equation (10) states that if product i is produced in period t, then the total amount produced up to period $t-1$ must meet the total demand up to period $t-1$. However, if the product is not made in period t, then the amount produced up to period $t-1$ must meet the demand up to period t. From our observation, this equation reduces the relaxation gap significantly and helps reduce the number of nodes branched on in a branch-and-bound solution method. Finally, equation (11) represents the labor constraints that link the problems across all products.

Due to the modelling of discontinuous integer values ($x_{it} \in \{0, E_i, E_i + L_i \dots\}$) for items $i \in SKU_2$ with binary variables z_{it}, solving large problems is extremely expensive. We therefore attempted a Lagrangean relaxation technique (see (Beasley 1993) for an overview of Lagrangean relaxation) where the problem is decomposed by relaxing the equations (11) to obtain the value of binary variables and then solving problem (P) for fixed value of binary variables, thereby solving subproblems that are less expensive to solve in each step.

Lagrangean Relaxation Approach

The Lagrangean relaxation method used for solving the problem (P) relaxes the complicating constraints (11) using Lagrange multipliers, thus resulting in a relaxed problem that is decomposable for each i. The relaxed problem (PL) is as follows

$$\textbf{PL:} \min_{x_{it}, y_{it}, z_{it}, s_{it}} [\sum_{i=1}^{N} \sum_{t=1}^{T} C_i s_{it}] - \sum_{t=1}^{T} \lambda_t \sum_{i=1}^{N} (R_i x_{it} - T_t)$$

subject to Equations (5)-(10).

Thus, (PL) is a relaxation of (P) and represents a lower bound to the solution of (P). Since (PL) is decomposable with respect to i, each subproblem is combinatorially less complex, and can be solved to determine the variables z_{it}. Then, for fixed values of z_{it}, the problem (P) may be solved to determine a specific solution that is an upper bound to the solution of (P). We note that due to the discrete lot-sizes, the integer solution of (P) may result in slacks in equation (11) and therefore may result in all multipliers of value zero (to satisfy complementary slackness). Therefore, the multipliers λ_t for the next iteration were obtained from the LP relaxation of (P). The problem is then solved iteratively until the bounds converge. Note that the bounds are not guaranteed to converge as there may be a duality gap due to discrete nature of the problem.

Restricting the Problem: $L_i = E_i$

It is comparatively easier to solve the problem when x_{it} has no discontinuous discrete integer values. Thus, with the assumption $L_i := E_i \;\; \forall i \in SKU_2$, binary variables z_{it} and equations (8) and (10) can be eliminated from the formulation. Restricting a given problem instance increases the lot-sizes for all products in SKU_2, thereby reducing the set of feasible solutions. As we could not find solutions to the unrestricted problem with CPLEX, we used restricted models for all experiments with IP branch-and-bound. The restricted problem is a sub-class of the original problem.

Experimental Results

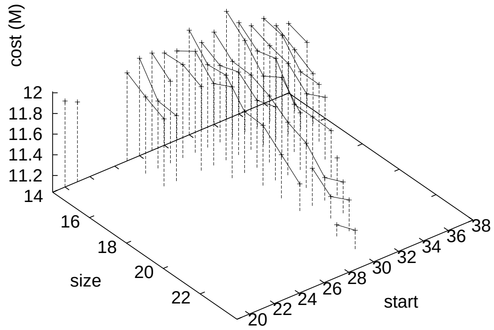
The experimental results reported in this section are based on a study of real data for 190 items and 52 weeks provided by a client of i2 Technologies from the process industry. The OIP model resulting from the given data is large: 7520 finite domain variables (average domain size 10) and 3047 constraints (average number of variables 30, 1525 constraints soft). To summarize the experimental results from the viewpoint of the client, what has the study achieved? (i) It found a solution which is provably within 1.4% of the optimal total cost for constant labor (two shifts), which (ii) shows that substantially cutting down cost requires reducing labor. (iii) It showed that labor can be reduced to one shift in up to 25 weeks with over 15% potential savings of total cost (or USD 1.9 million).

Comparison of Solvers

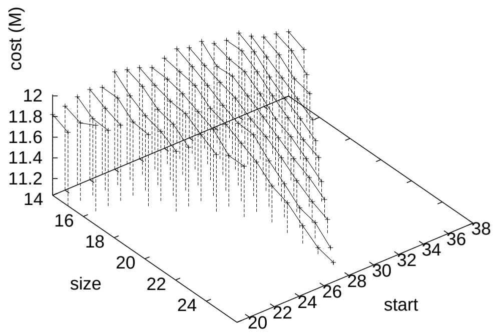
Table 3 reports the best solutions found by CPLEX and WSAT (OIP) in limited time and for different labor profiles. The table divides horizontally and vertically, distinguishing the original from the restricted model and the fixed-capacity from the varied-capacity case.

cost	real problem WSAT (OIP)	restricted problem CPLEX	restricted problem WSAT (OIP)
profile	fixed capacity, two shifts (230K)		
labor	11,960,000	11,960,000	11,960,000
inventory	1,023,106	1,120,680	1,040,373
total	12,983,106	13,080,680	13,000,373
profile	one shift [28,52]	one shift [30,51]	
labor	9,085,000	9,430,000	9,430,000
inventory	1,961,049	1,715,043	1,819,634
total	11,046,049	11,145,043	11,249,634

Table 3: Computational results with WSAT (OIP) and CPLEX 4.0. The restricted model forces $L_i := E_i$.



(a) CPLEX on the restricted MILP model.



(b) WSAT (OIP) on the over-constrained IP model.

Figure 2: Solutions for various labor profiles. Each impulse represents the total cost of the best solution found at one labor profile (start/size coordinates correspond to profiles $[\text{start}, \text{start+size}-1]$, the vertical axis is overall cost).

With respect to overall quality, the best solutions among all profiles obtained from both methods are approximately equal (WSAT (OIP) leading by less than 1% of the total cost, or USD 98,994). In the experiments, the runtime of WSAT (OIP) was limited to 10 minutes, CPLEX and was allowed 15 minutes for optimization and was cut-off after 30 minutes in case no feasible solution was found. All experiments were performed on a Sun Sparc Ultra II. Run-times were kept short because many labor profiles had to be examined to find solutions of good overall quality.

Figure 2 visualizes the experiments across different labor profiles. The right edge of the triangle reflects the fact that the size of the one-shift period must decrease as week 52 is approached, because the planning horizon is finite. On the restricted model, CPLEX could not find a solution with more than 22 one-shift periods in the given time while WSAT (OIP) was able to solve a problem with 25 one-shift periods. In general, CPLEX had difficulties to find feasible solutions as the labor constraints were tightened: Of 115 profiles solved by WSAT (OIP), CPLEX only solved 68 profiles (59%) within the given time limit (for comparison, WSAT (OIP) could still solve 100 given the restricted model). For the profiles that could be solved with both methods, WSAT (OIP) found better solutions in 41 cases; CPLEX found better solutions in 22 cases, despite the fact that it was applied to the restricted model. In the cases where WSAT (OIP) [CPLEX] was better, on average it improved over CPLEX [WSAT (OIP)] by 3.7% [1.3%] with respect to pure inventory cost.

Parameters CPLEX was run with standard parameter settings. In all experiments with WSAT (OIP), the following parameters were used: Initial production was set to zero ($p_{\text{zero}} = 1$), and a number of 10 tries were performed, each with 100K moves. Allowed variable triggers were limited to 2 steps up or down the current variable value. Hard constraints were repaired with high priority ($p_{\text{hard}} = 0.9$). Random moves appeared to deteriorate the solution quality, therefore we set $p_{\text{noise}} = 0$. A long tabu tenure appeared to be important to find feasible solutions for problems with very tight capacity ($t = 100$). Constraint weights were critical to obtain good feasible solutions and were assigned statically: The hard NOH constraints were weighted with a large number, expressing a preference to keep NOH constraints satisfied. In contrast, CAP constraints were weighted below 1.0 so that temporarily violating them during the search was encouraged.

Lower Bounds

To assess the quality of the solutions, we applied bound reasoning based on Lagrangean relaxation as described above. We used a relaxed labor profile of constant 300K, which is over two shifts per week and therefore an unrealistic problem. For a precise estimate of the solution quality, table 4 reports pure inventory costs based on this profile for the different methods. Using Lagrangean decomposition, we found solutions to the relaxed labor profile, but unfortunately could not find solutions for realistic capacity constraints. Table 4 also indicates that WSAT (OIP) is still considerably away from the best Lagrangean relaxation based solution (3.4% of inventory costs). With respect to the overall cost of this profile, the difference vanishes (0.2%). The reported lower bound is valid also for the original problem with constant two-shift labor, because the 300K-problem is a relaxation of the original problem.

Solution/bound ($T_t = 300\text{K}$)	type	value
Best IP solution	restricted	986,780
Best solution from WSAT (OIP)	restricted	973,834
Best solution from WSAT (OIP)	original	942,511
Best Lagrangean solution	original	911,960
Best valid lower bound	original	839,875

Table 4: Solutions (inventory cost) based on a fixed-capacity labor profile of 300K in all weeks.

Conclusions

We have studied a real-world capacitated lot-sizing problem (CLSP) from the process industry. Because the problem includes discrete lot-size requirements not reported in the CLSP literature, existing domain-specific methods are not directly applicable. We therefore approached the problem with WSAT (OIP), a new domain-independent local search method for integer optimization. WSAT (OIP) operates on over-constrained integer programs and generalizes the WSAT $(\mathcal{PB})$ heuristic. We experimentally compared the results to a commercial mixed integer programming solver.

While exact techniques for general purpose integer optimization (such as IP branch-and-bound) are widely researched and developed into industrial tools (such as CPLEX), few domain-independent heuristics for combinatorial optimization have been described. In this paper, we have presented a new local search heuristic for integer optimization and evaluated its performance on a capacitated production planning problem. Although the research on integer local search (ILS) is only at its beginning, the empirical results are promising: Integer local search can solve a CLSP with discrete lot-sizes of which a commercial MIP solver can only solve a sub-class. In terms of robustness, WSAT (OIP) is superior to CPLEX on the given data, in particular as the capacity constraints are tightened. The ILS model is simpler than the MIP model, and with respect to solution quality, the techniques are on par.

Acknowledgments

The first author is supported by a DFG (Deutsche Forschungsgemeinschaft) doctoral fellowship. We would like to thank Jimi Crawford for guidance and support in the course of this study. We are indebted to i2 Technologies, Ravi Gujar, Srinivasan Kumar, and the client from the process industry for making the publication of this study possible. Many thanks to the optimization teams of i2 Technologies and to Alexander Bockmayr, Martin Müller, Joachim Niehren, Gert Smolka, and Jörg Würtz for valuable suggestions and comments.

References

Aboudi, R., and Jörnsten, K. 1994. Tabu search for general zero-one integer programs using the pivot and complement heuristic. *ORSA Journal on Computing* 6(1):82–93.

Abramson, D.; Dang, H.; and Krishnamoorthy, M. 1996. A comparison of two methods for solving 0–1 integer programs using a general purpose simulated annealing algorithm. *Annals of Operations Research* 63:129–150.

Balas, E., and Martin, C. 1980. Pivot and complement – a heuristic for zero-one programming. *Management Science* 26:86–96.

Beasley, J. E. 1993. Lagrangean relaxation. In Reeves, C. R., ed., *Modern Heuristic Techniques for Combinatorial Problems*. Halsted Press. 70–150.

Bitran, G., and Yanasse, H. 1982. Computational complexity of the capacitated lot size problem. *Management Science* 28:1174–1186.

Borning, A.; Freeman-Benson, B.; and Wilson, M. 1996. Constraint hierarchies. In Jampel, M. B.; Freuder, E.; and Maher, M., eds., *Over-constrained Systems*. Springer.

Connolly, D. 1992. General purpose simulated annealing. *Journal of the Operational Research Society* 43:495–505.

Diaby, M.; Bahl, H.; Karwan, M.; and Zionts, S. 1992. A Lagrangean relaxation approach for very-large-scale capacitated lot-sizing. *Management Science* 38(9):1329–1340.

Drexl, A., and Kimms, A. 1997. Lot sizing and scheduling – survey and extensions. *European Journal of Operational Research* 99:221–235.

Glover, F., and Laguna, M. 1993. Tabu search. In Reeves, C. R., ed., *Modern Heuristic Techniques for Combinatorial Problems*. Halsted Press. 70–150.

Glover, F., and Laguna, M. 1997. General purpose heuristics for integer programming–part I & II. *Journal of Heuristics* 2/3(4/2):343–358/161–179.

Hindi, K. 1996. Solving a CLSP by a tabu search heuristic. *Journal of the Operational Research Society* 47:151–161.

Kirca, Ö., and Kökten, M. 1994. A new heuristic approach for the multi-item dynamic lot sizing problem. *European Journal of Operational Research* 75:332–341.

Løkketangen, A.; Jörnsten, K.; and Storøy, S. 1994. Tabu search within a pivot and complement framework. *Int. Transactions on Operations Research* 1(3):305–316.

McAllester, D.; Selman, B.; and Kautz, H. 1997. Evidence for invariants in local search. In *Proceedings AAAI-97*.

Nemhauser, G., and Wolsey, L. 1988. *Integer and Combinatorial Optimization*. Series in Discrete Mathematics and Optimization. Wiley-Intersience.

Parkes, A., and Walser, J. 1996. Tuning local search for satisfiability testing. In *Proceedings AAAI-96*, 356–362.

Selman, B., and Kautz, H. 1993. Domain-independent extensions to GSAT: Solving large structured satisfiability problems. In *Proceedings of IJCAI-93*.

Selman, B.; Kautz, H.; and Cohen, B. 1994. Noise strategies for improving local search. In *Proceedings AAAI-94*, 337–343.

Walser, J. 1997. Solving linear pseudo-boolean constraint problems with local search. In *Proceedings AAAI-97*.