GRASP WITH PATH-RELINKING FOR THE MULTI-PLANT CAPACITATED LOT SIZING PROBLEM

MARIÁ C. V. NASCIMENTO, MAURICIO G.C. RESENDE, AND FRANKLINA M. B. TOLEDO

ABSTRACT. This paper addresses the independent multi-plant, multi-period, and multi-item capacitated lot sizing problem where transfers between the plants are allowed. This is an NP-hard combinatorial optimization problem and few solution methods have been proposed to solve it. We develop a GRASP (Greedy Randomized Adaptive Search Procedure) heuristic as well as a path-relinking intensification procedure to find cost-effective solutions for this problem. In addition to this, the paper addresses applications of the proposed heuristics for the special case of the capacitated lot sizing problem with parallel machines. The results of the computational tests show that the proposed heuristics outperform other heuristics previously described in the literature. The results are confirmed by statistical tests.

1. INTRODUCTION

The capacitated lot sizing problem (CLSP) addressed in this paper is a combinatorial optimization problem whose objective is to find a production plan that minimizes production, setup, and inventory costs, and meets without delay the demands of items in the periods. According to Karimi et al. (2003), the CLSP is one of the most important and difficult problems in production planning. For the case in which setup times are considered, the problem to find a feasible solution is NP-complete (Maes et al., 1991). The single plant problem has been studied widely (Trigeiro et al., 1989; Lozano et al., 1991; Diaby et al., 1992a;b; Armentano et al., 1999). Moreover, numerous surveys have been published (Bahl et al., 1987; Kuik et al., 1994; Wolsey, 1995; Karimi et al., 2003).

According to Bahl et al. (1987), one can classify lot sizing problems as single-stage (with one planning stage) or multi-stage (with several planning stages). A system has a single stage when the items to be produced are independent, i.e., one item does not depend on the other to be produced. On the other hand, a multi-stage system is characterized by the fact that production of each item generates dependent demand for its components, whose production or purchase should also be planned.

The CLSP with parallel machines consists of a limited number of sets of machines where any set can produce the same items in an environment composed of a single stage and one plant. The machines can have different production and setup costs, and can as well be capacitated. The differences between the CLSP with parallel machines and the problem focused in this paper are the transfer costs, which for the

Date: January 21, 2008.
Key words and phrases. Lot sizing, multi-plant, parallel machines, GRASP, path-relinking.
AT&T Labs Research Technical Report TD-7B46YC.

former are not taken into account, as well as the fact that the CLSP with parallel machines has its own warehouse, which implies a single inventory cost. The CLSP with parallel machines has been studied by Lasdon and Terjung (1971), Carreno (1990), and Toledo and Armentano (2006).

In this paper, we address the single-stage, multi-plant, multi-item, and multi-period capacitated lot sizing problem. The problem considers transfer costs among plants and individual per-period plant demands. These transfer costs are incurred because we allow a plant to produce items for another plant. Likewise, we allow storage of items in plants distinct from the one in which the item is produced and/or is demanded. Since customers only pay for transportation from the nearest plant to the delivery location, the eventual additional transfer costs must be accounted for. Since the problem to find a feasible solution to the single plant capacitated lot sizing problem with setup time is NP-complete, so is its multi-plant variant. As we show later, exact methods encounter difficulties to solve instances of moderate size. Therefore, the use of heuristics as solution methods for this problem is justified. Some applications of these problems can be found in diversified manufacturing sectors, for example, in the mattress, stainless steel, and beverage industries, where plants are spread out geographically.

Multi-plant lot sizing problems may be classified into one of two types. The *dependent* type are those whose plants need each other to produce items, i.e., the production environment has more than one stage and some items need other items from other plants to be produced (Bhatnagar et al., 1993; Wu and Golbasi, 2004; Kaminsky and Simchi-Levi, 2003). The *independent* type are those whose production centers are independent, i.e., the plants individually supply the items demanded (Sambasivan and Schimidt, 2002; Sambasivan and Yahya, 2005). In both cases, the transfer of lots within the plants are accounted for and the optimal solution to the problem involves production planning integrating the whole set of plants.

The focus of this paper is the single-stage independent multi-plant lot sizing problem (MPCLSP). Few papers have previously addressed this problem and few solution methods have been proposed. Sambasivan and Schimidt (2002) described a heuristic based on transfers of production lots. Although the authors described most of the parameters used in their computational tests, they are not clear in the definition of *loose* and *tight* capacities, which makes their experiments difficult to reproduce. Sambasivan and Yahya (2005) proposed a method based on Lagrangian relaxation. In computational tests, the authors observe that the mean gap of their solution with respect to the optimal is inversely proportional to the number of items. The configuration of instances in Sambasivan and Yahya (2005) was clear and we are able to reproduce their experiments on the same set of instances.

This paper proposes a greedy randomized adaptive search procedure (GRASP) heuristic embedded with a path-relinking strategy to find cost-effective solutions to the MPCLSP. The procedure for generating the initial solutions for the GRASP uses a greedy randomized version of the exact algorithm of Sung (1986) for the uncapacitated lot sizing problem with multiple machines. These initial solutions are usually infeasible, forcing us to apply transfer of lots between periods and plants to restore feasibility before applying the local search procedure. To analyze the performance of the heuristic, we designed three experiments. In the first experiment, we tested the heuristics using the instances proposed in Sambasivan and Yahya

(2005). In the second experiment, our heuristics are tested using randomly generated instances according to Toledo and Armentano (2006). In both experiments our results outperformed those of the literature. Finally, in the third experiment, we used the methodology proposed in Aiex et al. (2002) and Aiex et al. (2007) to assess experimentally the running time distributions of our randomized algorithms.

The paper is organized as follows. We begin by presenting the mathematical model in Section 2 and provide the algorithmic details in Section 3. Section 4 deals with the computational experiments. Finally, in Section 5, we conclude the paper with some suggestions for future research.

2. Mathematical formulation

We review a mathematical model for the MPCLSP, due to Sambasivan and Schimidt (2002). In this model, the terms $\forall i$, $\forall j$, and $\forall t$, indicate any element belonging to, respectively, sets NI, MI, and TI (which we describe below). This mixed-integer programming model is:

$$\min \sum_{i \in NI} \sum_{j \in MI} \sum_{t \in TI} [c_{ijt} x_{ijt} + s_{ijt} y_{ijt} + h_{ij} I_{ijt} + (\sum_{k \in MI, k \neq j} r_{jk} w_{ijkt})]$$

subject to:

$$(1) \quad I_{ijt-1} + x_{ijt} - \sum_{k \in MI, k \neq j} w_{ijkt} + \sum_{l \in MI, l \neq j} w_{iljt} - I_{ijt} = d_{ijt} \qquad \forall i, \forall j, \forall t$$

$$(2) \quad x_{ijt} \leq (\sum_{j \in MI} \sum_{l=t}^{T} d_{ijl}) y_{ijt} \qquad \forall i, \forall j, \forall t$$

$$(3) \quad \sum_{i \in NI} (b_{ijt} x_{ijt} + f_{ijt} y_{ijt}) \leq C_j \qquad \forall j, \forall t$$

$$(4) \quad I_{ij0} = 0 \qquad \forall i, \forall j$$

$$(5) \quad x_{ijt}, I_{ijt} \geq 0 \qquad \forall i, \forall j, \forall t$$

$$(6) \quad w_{ijkt} \geq 0 \qquad \forall i, \forall j, \forall k, \forall t$$

$$(7) \quad y_{ijt} \in \{0, 1\} \qquad \forall i, \forall j, \forall t$$

where T, N, and M are, respectively, the number of periods, items, and plants in the planning horizon, TI, NI, and MI are, respectively, the sets $\{1, \ldots, T\}$, $\{1, \ldots, N\}$, and $\{1, \ldots, M\}$, d_{ijt} is the demand of item i at plant j in period t, C_j is the available capacity of production at plant j, b_{ij} is the time to produce a unit of item i at plant j in period t, f_{ijt} is the setup time to produce item i at plant j in period t, c_{ijt} is the unit production cost of item i at plant j in period t, s_{ijt} is the setup cost of item i at plant j in period t, h_{ij} is the unit inventory cost of item i at plant j, r_{jk} is the unit minimum transfer cost of an item from plant j to k, decision variables x_{ijt}, I_{ijt}, and w_{ijkt} are, respectively, the quantity of item i produced at plant j in period t, stored at plant j at the end of the period t, and transferred from plant j to plant k during period t, and decision variable y_{ijt} is a binary variable that assumes value 1 if item i is produced at plant j in period t, and 0, otherwise.

The minimum transfer cost r_{jk} represents the minimum cost to transfer any item from plant j to plant k and satisfies the triangle inequality $r_{ij} + r_{lk} \geq r_{jk}$. The objective function encodes the goal of the optimization, which is the minimization of the total cost, i.e., production, setup, inventory, and transfer costs.

Constraints (1) refer to the inventory balance of the quantity of item i during period t at plant j. These constraints ensure that the demand of item i in period t at plant j is met by the production of this item in period t at plant j, added to the amount of the item stored in the previous period at the that plant and the quantity to be transferred from other plants to plant j, subtracted by the quantity of item i in period t that is transferred to the other plants and the quantity of item i that is stored in period t at plant j. Constraints (2) ensure that if item i is produced at plant j in period t, i.e., if $x_{ijt} > 0$, then the binary variable $y_{ijt} = 1$, which implies that the setup of the plant is to be considered. Constraints (3) ensure that the available capacity is not violated, while constraints (4) impose empty initial inventories. Finally, constraints (5–7) impose the non-negativity of variables x, I, and w, and ensure that y is binary.

The differences between the mathematical models of the MPCLSP and the CLSP with parallel machines are the non-attendance of transfer costs in the objective function and the presence of a single warehouse in the CLSP model, which implies the unique inventory cost and production center.

3. Solution method

Metaheuristics are high-level procedures specialized to solve combinatorial optimization problems. They guide other simpler heuristics to search for good-quality feasible solutions. To find approximate solutions to the MPCLSP, we proposed two heuristics, a pure GRASP and a GRASP with path-relinking. GRASP was first proposed by Feo and Resende (1989) and Feo and Resende (1995). See also Resende and Ribeiro (2002) for a recent survey and Festa and Resende (2002) for a survey of a wide range of successful applications of GRASP.

GRASP is a metaheuristic based on a multi-start strategy, i.e., many initial solutions are generated through repeated applications of a semi-greedy process. Local search is applied at each multi-start iteration starting from the semi-greedy solution in an attempt to improve the quality of the constructed solution. The semi-greedy process builds a solution, one element at a time. At each step of this construction all candidate elements, i.e., elements whose inclusion in the solution do not lead to infeasibility, are analyzed with respect to their contribution to the cost of the solution. A restricted candidate list (RCL) with some of the best-valued candidates is set up. The construction procedure selects a candidate element from the RCL at random and adds this element to the solution under construction. This procedure is repeated until a feasible solution is on hand, i.e., until there are no more candidate elements to choose from.

In addition to the construction and local search components found in GRASP, a path-relinking intensification strategy is also incorporated in the GRASP with path-relinking heuristic. Path-relinking was originally proposed by Glover (1998) in the context of tabu search and scatter search. Its hybridization with GRASP was first proposed by Laguna and Martí (1999). See Resende and Ribeiro (2005) for a recent survey on GRASP with path-relinking. The hybridization of path-relinking with GRASP consists in the collection of a set of high-quality solutions, called elite solutions, found during the search. New solutions are developed by exploring trajectories that connect elite solutions and those produced by GRASP.

The GRASP with path-relinking proposed in this paper consists of multiple iterations of the following steps:

(1) Construct an initial solution with a semi-greedy procedure;
(2) If possible, make the solution feasible;
(3) Apply local search starting from the feasible solution;
(4) Apply the path-relinking strategy between the locally optimal solution found in the local search and some elite solution previously found.

The pure GRASP heuristic does not have the path-relinking phase.

In what follows, we describe both heuristics, by considering four phases. The first three phases make up the GRASP, while the last phase is the path-relinking intensification. These four phases are described in detail next.

3.1. **Initial solution.** Since finding a feasible solution for the multi-plant single period capacitated lot sizing problem is difficult, we propose to obtain an initial solution for the problem by relaxing the capacity constraints (3). Furthermore, if we ignore the transfer costs, the problem can be decomposed into n uncapacitated lot sizing problems on a parallel machine, where we associate each of the n items with each problem. This problem can be solved by the optimal algorithm of Sung (1986) which we describe below.

For each item i in the planning horizon, let Γ_{ijkt} be the production cost of item i at plant j in period $k+1$, to meet the demand d'_{ikt} of item i from period $k+1$ to period t for all plants $(1,\ldots,M)$, i.e.,

$$d'_{ikt} = \sum_{r=k+1}^{t} \sum_{j=1}^{M} d_{ijr},$$

with $k < t$, for $t = 1,\ldots,T$. Hence,

$$\Gamma_{ijkt} = \begin{cases} s_{ij} + c_{ij} d'_{ikt} + \sum_{r=k+1}^{t-1} h_{ij} d'_{irt} & \text{if } d'_{ikt} > 0, \\ 0 & \text{if } d'_{ikt} = 0. \end{cases} \tag{8}$$

Let ζ_{it} be the minimum production cost from period 1 to period t of item i, with an empty initial inventory. Therefore, ζ_{it} may be determined recursively by

$$\zeta_{it} = \begin{cases} \min\limits_{\substack{0\le k<t \\ 1\le j\le M}} \{\zeta_{ik} + \Gamma_{ijkt}\}, & \text{for } t = 1,\ldots,T, \\ 0, & \text{for } t = 0. \end{cases} \tag{9}$$

The dynamic programming forward recursion (9) is equivalent to the problem of finding a minimum cost path in an appropriately defined network. In this network, each node corresponds to a period of time and arc (k,t) is associated with the production of demand d'_{ikt} with cost Γ_{ijkt}. This problem can be solved by an efficient algorithm based on Evans (1985) and proposed by Armentano and Toledo (1997) with complexity $O(mT^2)$.

We produce the initial solution of the heuristic in the construction phase using the network corresponding to the uncapacitated lot sizing problem with parallel machines. This construction, for each item $i = 1,\ldots,N$, and specific period $t = 1,\ldots,T$, is done in the following four steps:

(1) Let S be the set of all possible paths from the period 1 node to the period t node, i.e., with all the possible values of ζ_{it}, and let the cost $g(j,s) = \zeta_{is} + \Gamma_{ijst}$, with $1 \le s \le t$ and $1 \le j \le M$.

(2) Define *gmin* and *gmax* as the minimum and maximum cost of $g(j,s)$, respectively.
(3) Let RCL be a restricted candidate list composed of the elements $s \in S$ such that j exists with $gmin \leq g(j,s) \leq gmin + \alpha(gmax - gmin)$, such that $\alpha \in [0,1]$.
(4) Select an element from the RCL at random and place it in the solution being constructed.

For each period we store, in list S, all possible paths in the network in increasing order of the $g(j,s)$ values. Considering the first element value in the list costs *gmin* and the last costs *gmax*, we insert only paths whose $g(j,s)$ values are less than or equal to $gmin + \alpha(gmax - gmin)$ in a second list called restricted candidate list (RCL). We select at random one such path from the RCL and continue building the solution until the last period.

This routine should be carried out for all items of the problem to obtain the initial solution which is, in most cases, infeasible. Feasibility must be restored before the local search procedure can be applied.

3.2. **Feasibility phase.** The feasibility procedure consists in transferring the production of items between periods and plants, inspired by the idea of Gopalakrishnan et al. (2001). The value of an infeasible solution is defined to be the sum of the objective function value of the solution and the total overtime capacity multiplied by an integer p.

To repair the constructed solution and make it feasible, we follow the six step procedure below:

(1) Search for the period and plant with the largest overtime capacity, and define this period-plant pair as the transfer *origination*;
(2) If there are no period and plant with overtime capacity use, then a feasible solution is found. Return it and stop.
(3) Determine, among all items, periods, and plants, which triplet {item, period, and plant} will be the transfer *destination*. To do this, consider all possible lot sizes to be transferred from the following: the maximum amount from the transfer origination (this cannot be in excess of what is produced in that period); the amount that cancels overtime from the transfer origination; or maximum quantity that the time destination allows. Among those, select the one that results in the best savings considering the origination transfer as defined in Step 1
(4) If the best savings is positive, execute the transfer and go back to Step 1;
(5) If the best savings is not positive, then do not further consider this period and plant in the evaluation until a positive savings has been found and go back to Step 1;
(6) If there are no period and plant without overtime capacity use, then the procedure cannot find a feasible solution, so stop.

In addition to the stopping criteria described above, we also limit the number of iterations.

3.3. **Local search procedure.** This local search procedure is similar to the repair routine. It differs in that we assume that the solution has been repaired and hence the periods and plants do not violate the capacity constraints. The local search can be summarized by the following three steps:

(1) Evaluate among all items, periods, and plants, which triplet {item, period, and plant} will be the origination and destination of the transfer and determine the amount of production to transfer. This amount will be either the amount of the item produced in the period at the plant of origination of transfer or the amount of the item that uses up only the available capacity at the plant and period of destination of transfer without causing overtime. Choose the triplets and amounts that result in the best savings;
(2) If the best savings is positive, execute the transfer and go to Step 1;
(3) If the best savings is zero, then return this solution.

3.4. **Path-relinking phase.** Path-relinking is a strategy which integrates intensification and diversification in search. It explores trajectories in the solution space connecting good-quality solutions (Glover, 1998). The solutions found in this path can be better than those being connected.

In this paper, we use a path-relinking strategy that is hybridized with the GRASP. This procedure maintains a fixed number of elite solutions found during the search which are combined with the GRASP solutions to produce perhaps better solutions. The combinations have a hierarchical choice, i.e., the new solutions are always built combining the configuration of each item of the lower value solution in the higher value solution. These combinations are summarized in five steps as follows:

(1) Build a new solution from the higher valued solution using the item configuration from the lower value solution which results in the best value solution while maintaining fixed the other item configurations from the higher value solution.
(2) Keep the resulting new solution.
(3) If the new solution is infeasible, then apply the feasibility phase of Section 3.2, and on the resulting feasible solution apply the local search of Section 3.3.
(4) If the resulting solution is better than the best solution obtained, update the incumbent.
(5) Go on obtaining the next item configuration from the lower value solution which results in the best value solution, substitute such a configuration in the solution kept in Step 2 and go back to Step 2.

The number of iterations of the above procedure is the number of items in the planning horizon.

4. Computational tests

We implemented the heuristics in C and carried out the computational experiments on an AMD Athlon 64 microcomputer, with 1GB of RAM and under Windows NT Operating System. The codes were compiled using the Borland C++ v. 6 compiler.

Computational tests involve three experiments. In the first, the heuristics are tested using instances of Sambasivan and Yahya (2005). We compared the solutions obtained with the results reported by Sambasivan and Yahya (2005). In the second experiment, the heuristics are tested using instances randomly generated according to Toledo and Armentano (2006). The quality of each solution is evaluated against the lower bound generated by a linear programming relaxation, since the solver cannot find the optimal solution for these instances in an acceptable computational time. In both experiments, for each instance both heuristics are applied ten times to check for robustness relative to the construction phase. The third experiment evaluates the probability distribution of the running time of the heuristics. Using time-to-target plots, their differences with respect to performance become very clear.

4.1. **Experiment I.** In this first experiment, we used the instances of Sambasivan and Yahya (2005) which consist of all combinations of 3, 4, 5, and 6 periods with 5, 10, and 15 items and with 3 and 4 plants. Each combination, or class, of instances were made up of 5 elements.

To evaluate the solutions, the percentage gap with respect to the optimal solution of the linear programming relaxation is computed as

$$Gap = \frac{(z_h - z_l)}{z_l} \times 100, \tag{10}$$

where z_h is the objective function value of the heuristic solution and z_l is the value of linear programming relaxation.

We define the classes of instances as: "Number of plants" × "Number of periods" × "Number of items." For example, the class made up of 3 plants, 4 periods, and 10 items is referred to as $3 \times 4 \times 10$. In all, there are 24 groups. The number of initial solutions and maximum number of iterations in the feasibility phase were set to 1000 and 100, respectively. The number of elite solutions we considered in this case was 15.

The experiments show that the proposed heuristics are robust, since the mean, worst, and best gaps are similar. For this reason, only the mean gaps are presented in the tables. In Table 1, we called `LRA` the heuristic of Sambasivan and Yahya (2005), `Gheur` corresponds the pure GRASP, while `GPRheur` is the GRASP with path-relinking. The columns labeled MG and MT indicate, respectively, the percentage mean gap and the mean time in seconds of each class of instances repeated ten times.

Both the Lagrangian relaxation heuristic (`LRA`) and the two GRASP heuristics obtained feasible solutions for 100% of the instances. Notice that, in Table 1, the GRASP with path-relinking variant (`GPRheur`) had the largest number of groups with the best mean gap between its solution and that of the Lagrangian relaxation heuristic (`LRA`), totaling 22 best mean gaps (in bold in Table 1) compared to two for `LRA`, the mean gap for `GPRheur` for all classes was also the best: 6.7% compared to 9.7% for `LRA`. Furthermore, the mean time for the pure GRASP was less than that of `LRA` on 17 of the 24 problem classes.

The heuristic `Gheur`, without the path-relinking phase, was less efficient than `GPRheur`, but, nevertheless, it was still competitive with published results in the literature. The gap was equal to the gaps reported in literature (9.7%) and mean running times were smaller (13.7 seconds versus 18.3 seconds).

To confirm the better performance of the proposed heuristic, we carried out a paired t-Student's test comparing the mean gaps of each class of instances of Sambasivan and Yahya (2005) with the pure GRASP and GRASP with path-relinking proposed in this paper. These statistical tests indicated that `GPRheur`, i.e., GRASP with path-relinking, was *significantly better* than `LRA`. Furthermore, the statistical test also showed that the `LRA` was *not significantly better* than pure GRASP.

Table 1. Results for Experiment I. For each problem class, the table lists for each heuristic (Lagrangian relaxation, GRASP, and GRASP with path-relinking), the mean percentage gap (MG) and mean solution time (MT) in seconds. Best gaps are in boldface.

	`LRA`		`Gheur`		`GPRheur`	
Class	MG	MT	MG	MT	MG	MT
$3 \times 3 \times 5$	7.1	7.7	8.1	0.9	**6.5**	1.9
$3 \times 3 \times 10$	12.1	13.5	6.5	2.2	**5.1**	5.7
$3 \times 3 \times 15$	15.3	11.6	8.1	4.8	**5.7**	14.6
$3 \times 4 \times 5$	7.5	4.4	7.7	1.4	**6.2**	3.7
$3 \times 4 \times 10$	11.7	31.0	7.3	5.3	**5.2**	11.2
$3 \times 4 \times 15$	13.4	21.2	9.2	11.2	**5.8**	31.5
$3 \times 5 \times 5$	13.9	3.7	9.5	3.3	**7.3**	6.2
$3 \times 5 \times 10$	10.2	3.4	10.6	10.2	**7.2**	21.0
$3 \times 5 \times 15$	11.7	15.8	9.9	21.0	**6.9**	49.3
$3 \times 6 \times 5$	8.2	13.6	9.1	4.8	**7.0**	8.4
$3 \times 6 \times 10$	9.2	11.8	10.8	14.2	**7.7**	35.1
$3 \times 6 \times 15$	9.6	17.7	11.6	22.8	**7.3**	67.1
$4 \times 3 \times 5$	6.1	8.9	7.3	1.4	**6.0**	3.2
$4 \times 3 \times 10$	11.3	11.7	7.6	4.8	**5.6**	11.5
$4 \times 3 \times 15$	7.1	10.9	9.4	10.2	**6.4**	29.6
$4 \times 4 \times 5$	8.8	6.3	8.8	3.2	**6.9**	6.2
$4 \times 4 \times 10$	9.2	27.1	9.6	10.0	**6.3**	22.9
$4 \times 4 \times 15$	10.7	33.5	11.4	20.8	**7.0**	56.3
$4 \times 5 \times 5$	8.2	24.6	10.8	5.9	**8.1**	11.4
$4 \times 5 \times 10$	**7.1**	33.1	10.7	19.3	7.2	41.2
$4 \times 5 \times 15$	8.7	20.6	11.6	39.1	**6.7**	95.8
$4 \times 6 \times 5$	**7.1**	29.7	11.2	9.0	8.2	14.8
$4 \times 6 \times 10$	8.1	30.9	11.9	32.9	**7.6**	71.3
$4 \times 6 \times 15$	10.1	46.1	13.9	70.2	**8.2**	166.3
Means	9.7	18.3	9.7	13.7	**6.7**	32.8

All statistical tests were done using the R 2.3.0 software package and correspond to the Student's t-paired test for multi-plant with a significance level of 0.05.

Table 2. Statistical results for mean gaps.

Alternative hypothesis	t	df	p-value
MG of GPRheur < MG of LRA	-1.708	23	0.00524
MG of Gheur > MG of LRA	0.0398	23	0.4843

In Table 2, we report the statistical results for mean gaps. Columns *df* and t correspond, respectively, to the degrees of freedom and the value of t-Student. Furthermore, we denoted the Alternative Hypothesis, abbreviating mean gap as MG, e.g., the Alternative Hypothesis *MG of GPRheur* $<$ *MG of LRA* means that the difference between the mean gaps of `GPRheur` are less than the mean gaps of `LRA`.

4.2. **Experiment II.** In the second experiment, the heuristics were tested on a set of 960 instances with the same dimensions as Section 4.1, but randomly generated according to Toledo and Armentano (2006), where the authors studied the problem with parallel machines, i.e., not considering the transfer costs and presence of only one warehouse for inventory.

We generated transfer costs as in Sambasivan and Yahya (2005) with the same interval of inventory costs. Different demands of items for each plant were generated and we adapted the plant capacities. The parameters were generated as follows. Unit production cost (c_{ij}) is randomly generated in U[1.5, 2.5]. Low setup cost (s_{ij}) is randomly generated in U[5.0, 95.0] while high setup costs are obtained by multiplying low setup costs by 10. Unit inventory (h_{ij}) and transfer (r_{jk}) costs are randomly generated in U[0.2, 0.4] and U[0.2, 0.4], respectively. Unit processing (b_{ij}) and setup times (f_{ij}) are randomly generated in U[1.0, 5.0] and U[10.0, 50.0], respectively. Low setup times are generated in this interval and high setup times are obtained by multiplying low setup time by 1.5. Finally, demand (d_{ijt}) is randomly generated in U[0, 180].

For each period of planning horizon, the capacities $Cap[j]$ of each plant j were generated according to

$$Cap[j] = [\sum_{t=1}^{T}\sum_{i=1}^{N}\frac{(b_{ij}d_{ijt} + f_{ij})}{T}]. \tag{11}$$

Let the normal and tight capacities be the value of equation (11) multiplied by, respectively, 1.0 and 0.9. The instances were generated with distinct setup costs and time values according to the high and low criteria. To define such classes of distinct instances, consider the notation of its definition: Capacity/setup cost/setup time: (*N*) Normal; (*T*) Tight; (*L*) Low; (*H*) High. Then, for example, the class of instances with normal capacity, low setup cost, and high setup time is denoted by *NLH*. Observe that the normal option only exists for the capacity, and the high option does not exist for this parameter.

In Table 3, we present the mean gaps and mean times of some classes of instances, the $4\times6\times15$ configuration, when the CPLEX 7.5 solver was applied with a threshold run time of 30 minutes for each instance. As a result, all instance classes but one could not be solved within the established time. Furthermore, we could not obtain an optimal solution in less than 10 minutes. Such poor results for the optimal solver suggest that heuristics may be more appropriate for these problems.

To evaluate the solution of the proposed heuristics we used a lower bound generated by a linear programming relaxation in the percentage gap calculation, as in Section 4.1. The number of initial solutions and maximum number of iterations in the feasibility phase were set in this heuristic at 200 and 100, respectively. The number of elite solutions we considered in this case was 15.

We report these results in Table 4 whose columns indicate respectively, the results of the GRASP without path-relinking (`Gheur`) and the GRASP with path-relinking

Table 3. CPLEX tests of some instances. For each problem class, the table lists the the mean percentage gap (MG) and mean solution time (MT) in seconds.

Class	Mean Gap (%)	Mean Time (s)
NHL	8.8	1800.2
NHH	8.4	1800.3
THL	9.5	1800.2
THH	9.2	1800.2
NLL	0.1	757.0
NLH	0.1	629.6
TLL	0.1	1106.1
TLH	0.1	1182.6
Means	4.6	1359.6

(`GPRheur`). The Class column indicates the problem class, while FEA indicates the feasibility incidence in percentage of the corresponding strategies. Observe that the GRASP heuristic with and without the path-relinking phase has the same FEA because the modification has just been done in the improvement phase of the solution approach. The MG and MT columns indicate, respectively, the mean gap in percentage and the mean time in seconds of each class of instances (120 instances each class) executed 10 times.

For instances with low setup cost, the mean gaps for `Gheur` and `GPRheur` are similar to those in Experiment 1, which confirms the quality of the heuristics. For instances with a high setup cost, the mean gaps are worse than the low setup instances. This is probably due to the fact that the linear problem tends to have small values for variable y and for high setup costs this results in a poor lower bound.

The average capacity utilization for both proposed heuristics is similar. It is 85.2% and 78.8%, for low and high setup costs, respectively. The instances of tight capacity use on average 85.6% of capacity, while for the normal capacity this value is 79.1%. The setup time consumes at most 12.6% of the normal or tight capacity.

We also tested our heuristics using the instances generated by Toledo and Armentano (2006) for the parallel machines version of the problem. For this problem, the GRASP reached feasibility in 97% of cases, while the heuristic proposed by Toledo and Armentano (2006) obtained feasible solution in 97.6% of cases. For the pure GRASP, the mean percentage gap with respect to the lower bounds reported by Toledo and Armentano (2006) and the mean times in seconds, were, respectively, 11.1% and 9.9 seconds. The GRASP with path-relinking obtained, respectively, 8.6% and 14.8 seconds, while the Lagrangian relaxation of Toledo and Armentano (2006) obtained, respectively, 10.0% and 7.4 seconds.

4.3. **Experiment III.** According to Aiex et al. (2002), GRASP and GRASP with-relinking have running time to the optimal solution that are distributed according to a shifted exponential distribution. Time-to-target (TTT) plots can be used to compare stochastic local search procedures by comparing their running time

TABLE 4. Results for Experiment II. For each problem class, the table lists the percentage of instances for which a feasible solution was found and for GRASP and GRASP with path-relinking, the percentage mean gap and mean solution time in seconds.

		Gheur		GPRheur	
Class	FEA	MG	MT	MG	MT
NHL	99.1	27.7	6.0	27.0	3.2
NHH	100	26.7	6.0	25.8	3.8
THL	97.5	31.4	5.4	31.3	3.1
THH	97.5	30.3	5.8	28.4	4.7
Means	98.5	29.0	5.8	28.1	3.7

		Gheur		GPRheur	
Class	FEA	MG	MT	MG	MT
NLL	100	8.7	17.8	8.2	7.6
NLH	100	8.6	17.5	8.1	7.1
TLL	98.3	9.8	15.0	9.2	7.8
TLH	99.1	9.5	15.8	9.0	7.7
Means	99.4	9.1	16.5	8.6	7.5
Total means	98.9	19.1	11.2	18.3	5.6

distributions. A TTT plot is generated by independently running an algorithm several times and measuring the time it takes to find a solution at least as good as a given target solution.

We compared `Gheur` and `GPRheur` by producing TTT plots with 50 independent runs for the ten instances of *THL* with six periods and three plants, in which five instances have five items and five have 15 items. As targets we used values 0.9% above the best solution found by `Gheur`. Time-to-target plots (TTT plots) were produced using Aiex et al. (2007) and are presented in Figures 1 and 2. The figures clearly show that the running times of the GRASP with path-relinking variant were much smaller than those of the pure GRASP.

5. CONCLUSIONS

This paper addressed the multi-plant capacitated lot sizing problem (MPCLSP) and proposed new approaches using GRASP and path-relinking to find good-quality solutions for this problem. The initial solution of GRASP is usually not feasible, forcing us to transfer lots between periods and plants to obtain feasibility. For path-relinking, we proposed combinations of initial solutions with a guiding solution and the feasibility and local search phases at each step of the way.

FIGURE 1. TTT Plots for *THL* instances with 6 periods, 3 plants, and 5 items.

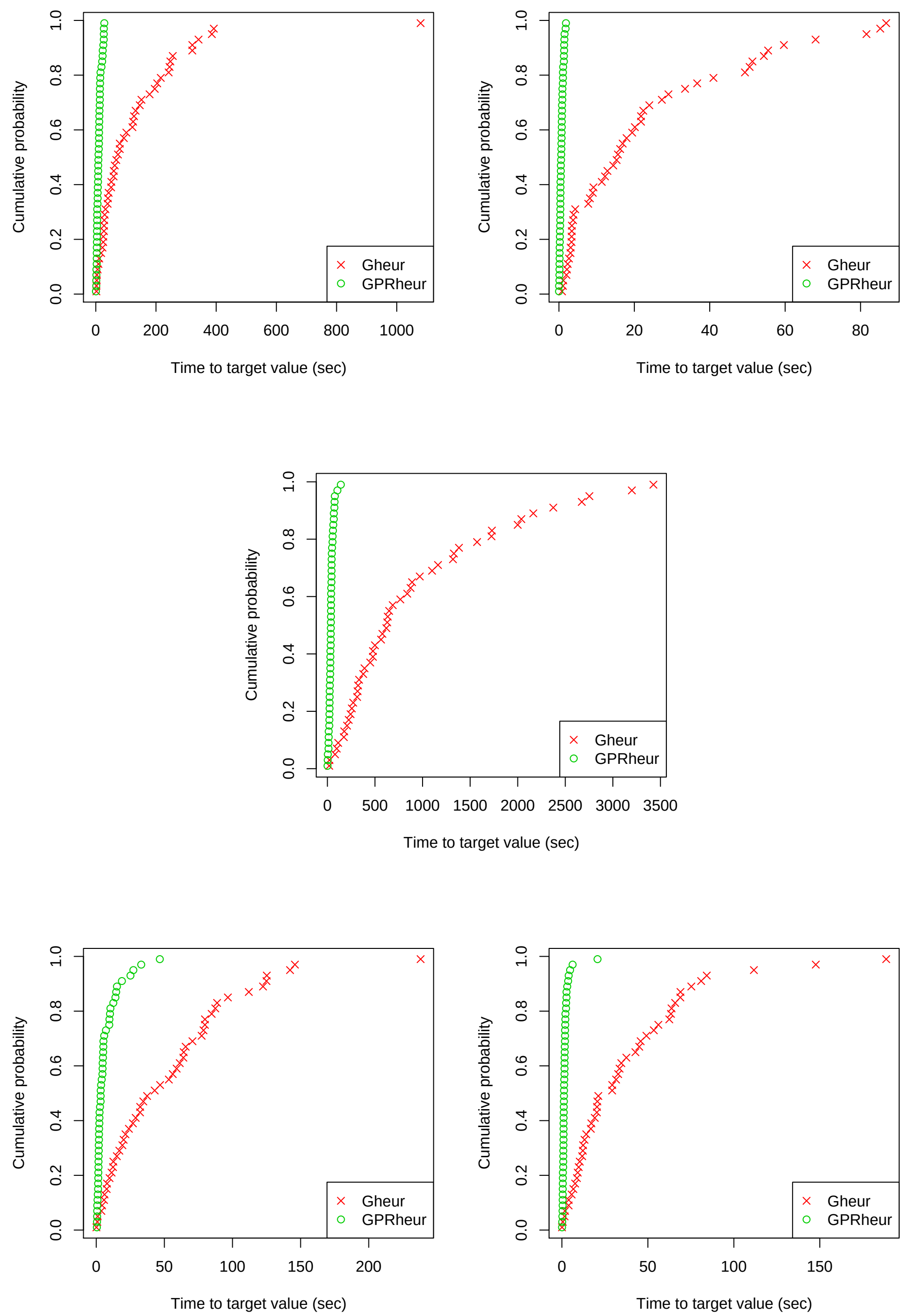
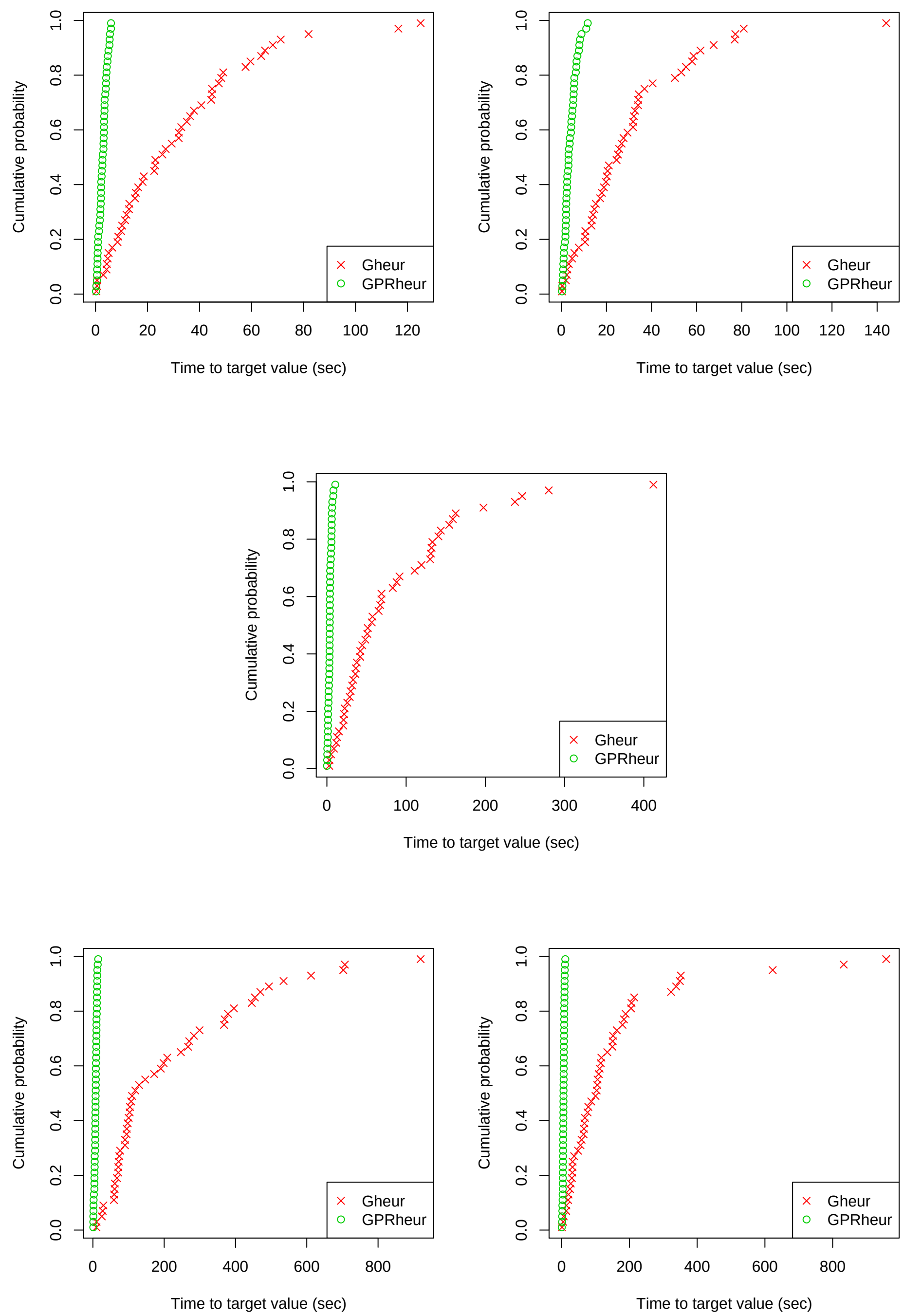


FIGURE 2. TTT Plots for *THL* instances with 6 periods, 3 plants, and 15 items.



The pure GRASP performed well for the MPCLSP when compared with the results in the literature. In some cases of multi-plant instances, the performance was better than the results presented in the literature. The mean solution times for our heuristics were also always better. When the strategy based on path-relinking was embedded into the GRASP heuristic, the results had an improved mean gap and performed better than heuristics in the literature. As a result, the mean gap of the GRASP with path-relinking was significantly better than that of the Lagrangian relaxation of Sambasivan and Yahya (2005) as statistical tests confirm. Thus, both GRASP heuristics proposed in this paper are new important solution approaches for the MPCLSP.

We also applied these heuristics to the parallel machine lot sizing problem and the results obtained are better than those in the literature. The heuristics can be adapted to other classes of lot sizing problems, such as single machine lot sizing problems and lot sizing problems with carry-over.

Acknowledgments

The authors are grateful to Mário de Castro Andrade Filho for the statistical assistance and to Murali Sambasivan for supplying the problem instances. This research was partially funded by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, Brazil).

References

R.M. Aiex, M.G.C. Resende, and C.C. Ribeiro. Probability distribution of solution time in grasp: An experimental investigation. *Journal of Heuristics*, 8:343–373, 2002.

R.M. Aiex, M.G.C. Resende, and C.C. Ribeiro. TTT plots: A perl program to create time-to-target plots. *Optimization Letters*, 1:355–366, 2007.

V.A. Armentano and F.M.B. Toledo. Dynamic programming algoritms for the parallel machine lot sizing problem. *Pesquisa Operacional*, 17:137–149, 1997.

V.A. Armentano, P.M. França, and F.M.B. de Toledo. A network flow model for the capacitated lot-sizing problem. *Omega*, 27:275–284, 1999.

H.C. Bahl, L.P. Ritzman, and J.N.D. Gupta. Determining lot sizes and resources requirements: a review. *Operations Research*, 35:329–345, 1987.

R. Bhatnagar, P. Chandra, and S.K. Goyal. Models for multi-plant coordination. *European Journal of Operational Research*, 67:141–160, 1993.

J. J. Carreno. Economic lot scheduling for multiple products on parallel identical processors. *Management Science*, 36:348–358, 1990.

M. Diaby, H.C. Bahl, M.H. Karwan, and S. Zionts. Capacitated lot-sizing and scheduling by lagrangean relaxation. *European Journal of Operational Research*, 59:444–458, 1992a.

M. Diaby, H.C. Bahl, M.H. Karwan, and S. Zionts. A lagrangean relaxation approach for very-large-scale capacitated lot-sizing. *Management Science*, 59:1329–1340, 1992b.

J. R. Evans. An efficient implementation of the wagner-whitin algorithm for dynamic lot-sizing. *Journal of Operational Management*, 5:229–235, 1985.

T.A. Feo and M.G.C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8:67–71, 1989.

T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

P. Festa and M.G.C. Resende. GRASP: An annotated bibliography. In C.C. Ribeiro and P. Hansen, editors, *Essays and Surveys on Metaheuristics*, pages 325–367. Kluwer Academic Publishers, 2002.

F. Glover. A template for scatter search and path relinking. In *AE '97: Selected Papers from the Third European Conference on Artificial Evolution*, pages 3–54, London, UK, 1998. Springer-Verlag. ISBN 3-540-64169-6.

M. Gopalakrishnan, K. Ding, J. M. Bourjolly, and S. Mohan. A tabu-search heuristic for the capacitated lot-sizing problem with set-up carryover. *Management Science*, 47:851–863, 2001.

P. Kaminsky and D. Simchi-Levi. Production and distribution lot sizing in a two stage supply chain. *IIE Transactions*, 35:1065–1075, 2003.

B. Karimi, S.M.T.F. Ghomi, and J.M. Wilson. The capacitated lot sizing problem: a review of models and algorithms. *OMEGA*, 31:365–378, 2003.

R. Kuik, M. Salomon, and L.N. Van Wassenhove. Batching decisions: structure and models. *European Journal of Operational Research*, 75:243–263, 1994.

M. Laguna and R. Martí. Grasp and path relinking for 2-layer straight line crossing minimization. *Informs Journal on Computing*, 11(1):44–52, 1999.

L. S. Lasdon and R. C. Terjung. An efficient algorithm for multi-item scheduling. *Operations Research*, 19:946–969, 1971.

S. Lozano, J. Larraneta, and L. Onieva. Primal-dual approach to the single level capacitated lot-sizing problem. *European Journal of Operational Research*, 51: 354–366, 1991.

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.

M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 219–249. Kluwer Academic Publishers, 2002.

M.G.C. Resende and C.C. Ribeiro. GRASP with path-relinking: Recent advances and applications. In T. Ibaraki, K. Nonobe, and M. Yagiura, editors, *Metaheuristics: Progress as Real Problem Solvers*, pages 29–63. Springer, 2005.

M. Sambasivan and C.P. Schimidt. A heuristic procedure for solving multi-plant, multi-item, multi-period capacited lot-sizing problems. *Asia Pacific Journal of Operational Research*, 19:87–105, 2002.

M. Sambasivan and S. Yahya. A lagrangean-based heuristic for multi-plant, multi-item, multi-period capacitated lot-sizing problems with inter-plant transfers. *Computers and Operations Research*, 32:537–555, 2005.

C.S. Sung. A single-product parallel-facilities production-planning model. *International Journal of Systems Science*, 17:983–989, 1986.

F.M.B. Toledo and V.A. Armentano. A lagrangean-based heuristic for the capacitated lot-sizing problem in parallel machines. *European Journal of Operational Research*, 175:1070–1083, 2006.

W.W. Trigeiro, L.J. Thomas, and J.O. McClain. Capacitated lot sizing with setup times. *Management Science*, 35:353–366, 1989.

L. A. Wolsey. Progress with single-item lot-sizing. *European Journal of Operational Research*, 86:395–401, 1995.

S.D. Wu and H. Golbasi. Multi-item, multi-facility supply chain planning: models, complexities, and algorithms. *Computational Optimization and Applications*, 28: 325–356, 2004.

(M.C.V. Nascimento) Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Caixa Postal 668, São Carlos,SP, CEP 13560-970, Brazil
E-mail address, M.C.V. Nascimento: mariah@icmc.usp.br

(M.G.C. Resende) Algorithms and Optimization Research Department, AT&T Labs Research, 180 Park Avenue, Room C241, Florham Park, NJ 07932 USA.
E-mail address, M.G.C. Resende: mgcr@research.att.com

(F.M.B. Toledo) Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Caixa Postal 668, São Carlos,SP, CEP 13560-970, Brazil
E-mail address, F.M.B. Toledo: fran@icmc.usp.br