INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

IFORS International Federation of Operational Research Societies

Intl. Trans. in Op. Res. 16 (2009) 173–187

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

A genetic algorithm for a bicriteria supplier selection problem

Chuda Basnet[a] and Andres Weintraub[b]

[a]*Waikato Management School, The University of Waikato, Hamilton, New Zealand,*
[b]*Department of Industrial Engineering, University of Chile, Santiago, Chile*
E-mails: chuda@waikato.ac.nz [Basnet]; aweintra@dii.uchile.cl [Weintraub]

Received 7 February 2008; received in revised form 2 October 2008; accepted 4 October 2008

Abstract

In this paper, we discuss the problem of selecting suppliers for an organisation, where a number of suppliers have made price offers for supply of items, but have limited capacity. Selecting the cheapest combination of suppliers is a straightforward matter, but purchasers often have a dual goal of lowering the number of suppliers they deal with. This second goal makes this issue a bicriteria problem – minimisation of cost and minimisation of the number of suppliers. We present a mixed integer programming (MIP) model for this scenario. Quality and delivery performance are modelled as constraints. Smaller instances of this model may be solved using an MIP solver, but large instances will require a heuristic. We present a multi-population genetic algorithm for generating Pareto-optimal solutions of the problem. The performance of this algorithm is compared against MIP solutions and Monte Carlo solutions.

Keywords: supplier selection; supplier rationalisation; bicriteria optimisation; genetic algorithm

1. Introduction

In the globalised business world of today, firms are increasingly focusing on their core activities and outsourcing the other activities. This has highlighted the methods used by firms in the selection of supply partners. While traditionally firms have simply chosen their suppliers on the basis of the lowest bids, the increased reliance on the suppliers has caused firms to choose their suppliers more carefully, on a long-term basis. To sustain and nurture these long-term relationships, firms are reducing the number of suppliers they deal with. The adoption of just-in-time philosophy and the emphasis on supply chain integration have also raised the need to reduce the number of suppliers. This has given rise to the twin goals of lowering the cost of supplies and of lowering the number of suppliers.

In this paper, we discuss the problem of selecting the suppliers for an organisation, where a number of suppliers have made price offers for supply of items, but have limited capacity. Selecting the cheapest combination of suppliers is a straightforward matter, but purchasers often

Published by Blackwell Publishing, 9600 Garsington Road, Oxford, OX4 2DQ, UK and 350 Main St, Malden, MA 02148, USA.

have a goal of lowering the number of suppliers they deal with. This second goal makes this issue a bicriteria problem – minimisation of cost and minimisation of the number of suppliers.

In the next section, a review of the literature is presented. Next, we discuss an optimisation model designed to address this problem. Algorithms to generate Pareto-optimal solutions of the model are discussed in the following section. Then we present the result of computational testing of the algorithms. We conclude with some final remarks.

2. Literature review

The greatest number of articles published in the area of supplier selection is qualitative – discussing the issues involved in the selection process in a normative or a descriptive fashion. Weber et al. (1991) have reviewed papers on supplier selection published since 1966 till 1991. They looked at 74 articles in the period and identified the selection criteria discussed in the articles. Their study was based on 23 supplier selection criteria identified earlier by Dickson (1966). Weber and colleagues tabulated the 74 articles, based on which of these 23 criteria were discussed, and in which year. In order of frequency, *net price*, *delivery*, and *quality* were the three most discussed criteria. They also reviewed the use of quantitative methods of vendor selection in these articles. The most frequently discussed method, by far, was the linear weighting scheme, where a weight is placed on some measure of the criterion, and the weighted measures are simply added. They found only 10 articles that used mathematical programming methodology. Of these, only two recognised the multi-criteria nature of the problem and applied techniques from the multi-objective programming (MOP) field. Consequently, Weber and colleagues suggest multi-objective programming as a fruitful area for future research in this area. We review below a few articles that adopt a quantitative orientation on supplier selection.

Buffa and Jackson (1983) provided an early model that recognised the multi-objective nature of vendor selection. Their model selected vendors and allocated orders based on the vendors' historical quality (acceptance rate), on-time arrival rates, late arrival rates, and early arrival rates. They modelled all the objectives as goals: minimisation of the sum of deviation from the goals was the objective of their linear programme. There were a total of five such goals in the objective function.

Narasimham (1983) illustrated the application of the analytic hierarchy process (AHP) to supplier selection. In their example, the top-level evaluation criteria were pricing structure, delivery, quality, and service. Following the AHP procedure, these criteria were further broken down into their components. In the next level, the vendors were compared pair-wise for their contribution to the above components. Final vendor rankings were obtained by summing up the weighting from the bottom-up. Barbarosoglu and Yazgac (1997) have presented a case study where the AHP procedure was used. Their model had a more extensive breakdown of the selection criteria than that of Narasimham (1983). Ghodsypour and O'Brien (1998) have extended the AHP approach by using the final AHP supplier scores themselves as weights for the supplier orders in a linear programme that seeks to maximise the sum of the products of supplier scores and supplier orders subject to capacity and quality constraints.

Weber and Current (1993) present a multi-objective, mixed integer programming (MIP) formulation of a single-item, single-plant vendor selection problem involving three criteria: price, quality, and delivery. Associated with each supplier, there is a price, a percentage defective, and a percentage late delivery. Limits on the maximum and minimum amount of business for particular suppliers, and on order quantity limits placed by the suppliers exist. The criterion of number of suppliers is modelled as a constraint. Standard integer programming (IP) software is used in conjunction with a weighting scheme to generate non-dominant solutions. Weber and Ellram (1993) present a decision support system designed to aid the purchasing manager using the above model as a basis. Weber et al. (2000) have extended this multi-objective model to include a data envelopment analysis to compare the efficiencies of employing different numbers of vendors.

Min (1994) has applied the multi-attribute utility approach of decision making to the supplier selection problem. This is essentially a scheme of weighing the multiple attribute with their scaling constants. Before this, each criteria score is converted to its utility function. The scaling and the utility function are elicited from the decision maker's trade-off of the multiple criteria.

Current and Weber (1994) have demonstrated how various supplier selection problems can be modelled as facility location problems. The "facility location" in facility location models is simply replaced by "supplier location" so that facility location models will work for supplier selection problems. Both models have the task of covering the customer demands. The authors have also suggested changes to the facility location models to make them amenable to address specific supplier selection considerations. Single objectives of cost minimisation or of supplier number minimisation are considered.

Jayaraman et al. (1999) presented a single-objective mathematical programming model that selects suppliers and allocates the orders to them. Their single objective includes the cost of the purchase and the cost of engaging a supplier. The model has constraints regarding lead time, quality, storage space, and production capacity of the suppliers.

Degraeve and Roodhooft (1999) presented a supplier selection and order allocation model for a single item over multiple time periods. This is a single-objective mathematical programming model that considers "total" costs in this scenario: supplier transaction costs, receiving costs, invoice costs, order costs, price discounts, set-up costs, etc. A case study involving the purchase of electrodes for a manufacturing company is discussed.

Karpak et al. (1999) discussed a case where the multi-objective technique of visual goal programming was used to address the supplier selection with the objectives of acquisition cost, quality, and delivery reliability.

Dahel (2003) considers three criteria: cost, percentage defectives, and percentage late deliveries, in a vendor selection model that includes the discount offered by suppliers for purchase volumes. This is a multi-objective mathematical programming model; however, the set of Pareto-optimal solutions is not explored: the model simply imputes penalty costs for defective items and late deliveries and thus converts the multiple objectives into a single objective. Computational times are presented for various problem sizes and numbers of discount brackets.

Although Buffa and Jackson (1983) modelled the supplier-selection problem as a multi-objective one, they effectively converted the objectives into one, by combining the deviations of all objectives from their goals using goal programming. Thus, they were able to use a straightforward linear program. Weber and Current (1993) maintained the separation of the objectives, but the size of their problem was small (they considered only one item, and the maximum number of suppliers is six in their example) and so they were able to use standard IP software to explore the non-dominant solutions. Karpak et al. (1999) used goal programming in their paper, as Buffa and

Jackson (1983) had done before. Dahel (2003) has used an MIP formulation and a commercial software to solve their model, but the maximum number of suppliers in their paper was 30; thus, the number of binary variables was not very high. Further, minimising the number of suppliers selected was not an objective in the paper.

In conclusion, the most common quantitative methodology presented in the literature is the linear weighting scheme. The next most common methodology is the AHP process. There are not many papers dealing with the application of mathematical optimisation to the supplier selection problem. Among them there are even fewer papers (Buffa and Jackson, 1983; Weber and Current, 1993; Karpak et al., 1999; Dahel, 2003) that explicitly model the problem as a multi-objective optimisation problem. All these papers consider the three objectives of the supplier selection process as cost, quality, and delivery; the number of suppliers is often modelled as a constraint rather than an objective. In this paper, we consider the number of selected suppliers as an independent objective in view of the importance of this objective in modern supply chain management.

The distinct contribution in our paper is the explicit recognition of the number of suppliers as an objective, presenting an MIP formulation, and designing a genetic algorithm to find the efficient frontier for larger-sized problems.

3. Bi-criteria supplier selection

Modern supply chain management emphasises close integration between suppliers and purchasers. Organisations are striving to have strategic relationships with suppliers. This calls for a reduction or a rationalisation of the number of suppliers. It is then possible to allocate scarce resources to develop relationships with the few chosen suppliers. In the earlier era of transactional relationships, it made sense to have as many suppliers as possible in order to find the cheapest deal. In these days of integrative relationships, reducing the number of suppliers has become one of the objectives of supplier selection. The model presented below has supplier reduction as one of the explicit objectives.

3.1. Problem statement

Notation
$i \in 1 \ldots n$, index of items
$j \in 1 \ldots m$, index of candidate suppliers
D_i = Demand for item i
p_{ij} = Price of supplier j to supply item i
C_{ij} = Capacity of supplier j to supply item i
q_{ij} = Quality (fraction defectives) of supplier j when supplying item i
Q_i = Acceptable quality for item i
l_{ij} = Delivery (fraction late) of supplier j when supplying item i
L_i = Acceptable delivery for item i
f_1 = Total cost of purchase
f_2 = Total number of suppliers selected

Decision variables
x_{ij} = Quantity of item i ordered to supplier j
y_j = 1 if supplier j is selected; 0 otherwise

Model
The optimisation problem in this paper is

$$\text{Min}\, Z = (f_1, f_2)$$

Subject to:

$$\sum_j x_{ij} \geqslant D_i, \quad i = 1, \ldots n. \tag{1}$$

All the item demands must be covered.

$$x_{ij} \leqslant C_{ij}, i = 1 \ldots n; \quad j = 1, \ldots m. \tag{2}$$

Capacity limits of each supplier may not be exceeded.

$$\sum_j x_{ij} q_{ij} \leqslant Q_i D_i, \quad i = 1, \ldots n. \tag{3}$$

Aggregate quality must be acceptable.

$$\sum_j x_{ij\, lij} \leqslant L_i D_i, \quad i = 1, \ldots . \, n. \tag{4}$$

Aggregate lateness must be acceptable.

$$x_{ij} \leqslant D_i y_j, \quad i = 1, n; j = 1, \ldots m. \tag{5}$$

Each selected supplier is counted.

$$f_1 = \sum_{ij} x_{ij} p_{ij}. \tag{6}$$

This calculates the total cost.

$$f_2 = \sum_j y_j. \tag{7}$$

This calculates the number of suppliers.

$$x_{ij} \geqslant 0; y_i \varepsilon (0, 1).$$

This is a mixed binary integer programming problem, with m binary variables (y_j), mn non-negative variables (x_{ij}), and two objectives (f_1, f_2). Quality and delivery are constrained to be within acceptable limits.

4. Solution approaches

Faced with multiple objectives, the analyst may opt for one of three approaches (Ehrgott and Gandibleux, 2002, p. 376). In the preference-oriented approach, the analyst has *a priori* knowledge of the decision maker's preferences so that the analyst can assign weightings to the different objectives. In this approach, the decision maker is presented with one solution that satisfies their preferences. Most of the multi-objective supplier selection models presented in the literature belong to this approach (Narasimham, 1983; Min, 1994; Dahel, 2003). Another approach is to let the preferences be articulated, as solutions are presented to the decision maker, interactively (Karpak et al., 1999). In the generating approach, the decision maker is presented with the complete set of non-dominated or Pareto-optimal solutions so that they can get a full perspective on the trade-offs involved. This is particularly suited to a problem with two objectives, where the trade-off between the two objectives can be clearly seen in a two-dimensional chart. This is the approach we have followed in this paper. We developed and tested the following algorithms for generating the set of Pareto-optimal solutions to the problem presented in the last section.

4.1. MIP

As mentioned above, the problem presented in this paper is a multi-objective MIP problem. It is possible to find the Pareto-optimal solutions by constraining the number of selected suppliers and solving the resulting MIP problem.

A pseudo code for such an algorithm is:

NumSuppliers = m;
While feasible solutions are found,
{ Min f_1
s.t.
constraints (1) to (6) above, and
$\sum_j y_j \leqslant$ NumSuppliers}
NumSuppliers = NumSuppliers − 1;
}

All the Pareto-optimal solutions to the problem will be generated by this MIP algorithm.

4.2. Multi-objective genetic algorithm (MGA)

Exact Pareto-optimal solutions to smaller- and medium-sized problems can be generated using the MIP formulation with an MIP solver, as discussed above. When the number of suppliers is high, MIP solvers are unable to generate all the Pareto-optimal solutions in a realistic time frame. Such larger-sized problems motivate the development of heuristic approaches.

The application of multi-objective programming to supplier selection is not very common, but the area of multi-objective programming itself is well established. Many books and articles have been published in this field. Jones et al. (2002) have reviewed the use of meta-heuristics in the area of multi-objective decision making. They noticed a significant growth in the use of meta-heuristics in the late 1990s. The primary area of application was within the multi-objective programming paradigm (as contrasted with goal programming or interactive methods). The most widely used meta-heuristic was genetic algorithms, followed by simulated annealing and tabu search. In this paper, we present a genetic algorithm for this problem.

Genetic algorithms start with a set of solutions, called *chromosomes*. In each iteration of the genetic algorithm, a selection procedure evaluates the chromosomes and selects a pair for *mating* and *mutation* in order to produce new offspring chromosomes. Thus, a new set (*populations*) of solutions is generated in each iteration. Genetic algorithms are especially suited for multi-objective optimisation because genetic algorithms work through generating multiple solutions, and multiple solutions are useful for the identification of non-inferior solutions in multi-objective optimisation.

In our genetic algorithm, each chromosome (solution) is a sub-set of the suppliers in our supplier set. Thus, each chromosome may be conceptualised as an array of size m, the elements in this array being binary integers – a "1" value signifying that the supplier is selected, a "0" value signifying that the supplier is not selected. In other words, each chromosome is a y_j vector. The genetic algorithm manipulates the chromosomes by mating and mutation to produce new chromosomes. Given a chromosome, one still needs to find the minimum (optimal) cost associated with it, thus evaluating the chromosome.

In applying the genetic algorithm to multi-objective programming, the issue of selecting the mating pair needs to be resolved. For single objectives, obviously the parents can be selected on the basis of their achievement of the single objective, but with multiple objectives all the various objectives need to be considered. Researchers have resolved this issue in different ways. For each selection instance, Murata et al. (1996) applied different random weights to the objectives to permit a wide selection of parents. Some other authors (e.g. Cochran et al., 2003) separated the populations into sub-populations, one for each criterion, and allowed the genetic algorithm to work separately for each population. Because in our case, one of the criteria (number of suppliers) is discrete, we separated the total population into sub-populations, based on the number of suppliers in the chromosome. From each of these sub-populations elite chromosomes are chosen, based on the cost, for mating and mutation. Because the offspring can have a different number of suppliers than the parents, the offspring are reclassified after each generation and inserted into the appropriate sub-population.

4.2.1. Evaluation of the chromosome

There are m suppliers, which could be included or excluded in any solution to this problem. Thus, the number of possible chromosomes is 2^m. Given a chromosome, which is a selected set J of suppliers, evaluating the chromosome entails allocating items to suppliers, which is a linear programming problem as given below.

$$\text{LP1 : Min} \sum\nolimits_i \sum\nolimits_{j\varepsilon J} x_{ij} p_{ij}$$

s.t.

$$\sum\nolimits_{j\varepsilon J} x_{ij} \geqslant D_i, i = 1, \ldots n. \tag{8}$$

All the item demands must be covered.

$$x_{ij} \leqslant C_{ij}, \quad i = 1 \ldots n, j\varepsilon J. \tag{9}$$

Capacity limits of each supplier in the set may not be exceeded.

$$\sum\nolimits_{j\varepsilon J} x_{ij} q_{ij} \leqslant Q_i D_i, \quad i = 1, \ldots . n. \tag{10}$$

Aggregate quality must be acceptable.

$$\sum_{j\varepsilon J} x_{ij\,lij} \leqslant L_i D_i, \quad i = 1, \ldots n. \tag{11}$$

Aggregate lateness must be acceptable.

$$x_{ij} \geqslant 0.$$

This linear program (LP1) needs to be solved for each chromosome generated by the genetic algorithm. However, the size of the linear program can be considerably reduced by making the observation that once a set of suppliers is selected, allocations for an item do not interact with allocations for any other item. Thus, LP1, above, can be broken down into n linear programs, one for each item i:

$$\text{LP2 : Min} \sum_{j\varepsilon J} x_{ij} p_{ij}$$

s.t.

$$\sum_{j\varepsilon J} x_i j \geqslant D_i. \tag{12}$$

All the demand for this item must be covered.

$$x_{ij} \leqslant C_{ij,j\varepsilon J}. \tag{13}$$

Capacity limits of each supplier in the set may not be exceeded.

$$\sum_{j\varepsilon J} x_{ij} q_{ij} \leqslant Q_i D_i. \tag{14}$$

Aggregate quality for this item must be acceptable.

$$\sum_{j\varepsilon J} x_{ij\,lij} \leqslant L_i D_i. \tag{15}$$

Aggregate lateness for this item must be acceptable.

$$x_{ij} \geqslant 0.$$

If one of these smaller linear programs is infeasible, the whole set J of suppliers (or the chromosome) is infeasible. LP2 minimises the cost of allocating items to the suppliers; this minimum cost provides the evaluation of the chromosome.

4.2.2. A heuristic bounding approach to reduce the number of LP2 iterations

For larger-sized problems, when the number of suppliers and number of items are high, evaluating each chromosome can take considerable time: n linear programs (LP2) need to be solved for each chromosome (which is a list of suppliers, $y_{j,}$ here), with $|y_j|$ variables and $|y_j|+3$ constraints. The number of chromosomes to evaluate can run into tens of thousands. We developed a fast heuristic confidence interval on the value (minimum cost) of a chromosome, so that only a promising chromosome is evaluated using linear programming. Such a lower bound on the optimum value of a chromosome is found by selecting suppliers from this list (y_j) in a greedy manner for each item, allocating demand to suppliers in a non-decreasing order of item-prices (picking the cheapest suppliers first), until the demand is met. If the demand cannot be met, obviously the chromosome is infeasible, and it does not need to be evaluated further. If the demand can be met, the chromosome may still be infeasible because of quality and delivery constraints. However, the price obtained by the above greedy allocation of suppliers provides a "greedy" lower bound (GLB) on the minimum price of a chromosome. Further, we examined the ratio of the minimal-cost (LP2) value of individual chromosomes to their lower bound (GLB). The variation of this bound ratio was found to be small. Hence, a statistical approach was followed to heuristically tighten the GLB. We evaluated the first 100 chromosomes of each problem by both the linear program (LP2) and the GLB to determine the average and standard deviation of the bound ratio. This gave us good statistical estimators for the variation of the bound ratio. These estimators were then used to find the 95% confidence estimate (CE) for the lower value of the bound ratio. Multiplying the GLB of the subsequent chromosomes by this estimate (CE) provided a tight heuristic lower bound (TLB) on the value of the chromosomes. The genetic algorithm then goes through the LP2 procedure to evaluate a chromosome only if this tightened lower bound (TLB) is lower than the lowest cost in the sub-population found so far; otherwise, the TLB is used as the evaluation. We found that this heuristic bounding approach reduces the use of the LP2 procedure significantly, using LP2 only in 20–30% of the evaluations. The solution found by the heuristically bounded procedure was within 1% of the solution found by using only the LP2 procedure.

4.2.3. Particulars of the genetic algorithm

The details of the genetic algorithm are given in this section. As mentioned before, the core of the genetic algorithm consists of a set of solutions, which are called chromosomes. In each iteration of the algorithm, the current set of chromosomes gives rise to a new set of chromosomes using the processes of *mating* and *mutation*. The iterations continue until a pre-set limit on the number of iterations.

- Chromosomes: in our algorithm, as described before, each chromosome is a vector y_j. Recall that if $y_j = 1$, the supplier j is selected; otherwise, supplier j is not in the solution.
- Evaluation: each chromosome is evaluated using the bounding procedure given in the last section. Traditionally, genetic algorithms work on maximisation problems. The current problem, however, is a minimisation problem. The evaluation turned the problem into a maximisation problem by subtracting the objective function from $\sum_{i \varepsilon \boldsymbol{I}} \boldsymbol{D_i}\, P_i$, where $P_{i = \mathrm{Max}\; j \varepsilon \boldsymbol{J}(p_{ij})}$. Thus, an evaluation (so-called *fitness*) represents the savings by using the current chromosome from the cost incurred by using the suppliers with the highest prices. When a chromosome is infeasible, a fitness of zero is returned.
- Initial population: initially, chromosomes are generated randomly, with the probability of selection of a supplier as 0.5. The number of initial chromosomes generated is $4 \times m$. Once a chromosome is generated, it is evaluated as above and placed in the sub-population corresponding to the number of selected suppliers.
- Selection: from each sub-population, two parent chromosomes are selected, the probability of selection of a parent being proportional to its fitness.
- Mating by crossover: two child chromosomes are generated from the two selected parent chromosomes by the crossover operation. The crossover site is generated randomly. The processes of crossover and mutation in the mating process can be illustrated using strings of binary digits. Each string is a chromosome, a set of suppliers (1 = selected; 0 = not selected).

Parent 1: $010\underline{10101}$
Parent 2: $111\underline{00110}$
Crossing over the entire string from fourth bit (crossover site) onwards from parent 1 to child 2, and parent 2 to child 1, while retaining the first 3 bits from parent 1 to child 1 and parent 2 to child 2 results in the following:
Child 1: $010\underline{00110}$
Child 2: $111\underline{10101}$

- Mating by crossover and mutation: two more parent chromosomes are selected from each sub-population. They are mated as above, and *mutated.* For example, mutating (randomly) bit 1 and 5 of above child 1 and bit 3 and 7 of child 2 (from 0 to 1 and vice versa) results in the children:
  Child 1: 11001110
  Child 2: 11010111
- The probability of mutation was set at 0.1.
- New population: all the child chromosomes are evaluated and placed in the sub-population corresponding to their number of suppliers. From each sub-population, four chromosomes with the highest fitness are retained, and others are discarded. The population is ready for new generation of chromosomes again.
- Iteration limit: the generation of new populations continued until the number of evaluations reached an evaluation limit, which was set at $150 \times m$.
- Solution: the final chromosomes in each sub-population are the four best solutions for the corresponding number of suppliers.

4.2.4. Monte Carlo optimisation (MCO)

A simple variation of the genetic algorithm described above would be to skip the mating and mutation operations and generate individual chromosomes directly. An MCO algorithm was implemented where supplier sets were generated by including (or excluding) a supplier in a set randomly (with a probability of 0.5). This way a list of suppliers, y_j, is generated each time. Items are allocated to suppliers in the list using the same bounded LP procedure as above. A list of non-dominated solutions found this way is maintained. This MCO algorithm was used to evaluate the performance of the MGA algorithm.

5. Computational results

To gauge the performance of the above algorithms, multiple problems were randomly generated and solved.

5.1. Problem generation

The number of suppliers was used as an indicator of the problem size. The number of items was set at three levels: half the number of suppliers, equal to the number of suppliers, and double the number of suppliers. The demand for an item was sampled from a uniform distribution of [1, 100].

The price of an item for any supplier was drawn from a uniform distribution of [1, 10]. The capacity of a supplier to supply an item was drawn from a uniform distribution of [1, 50]. If the total capacity of all the suppliers for an item was <2.5 times the demand for that item, their capacities were multiplied by 1.5 until the total capacity was at least 2.5 times the demand. The quality and delivery of a supplier for any item were each drawn separately from a uniform distribution of [0.01, 0.05]. The acceptable levels of quality and delivery were each set at 0.03.

5.2. Results

The MIP algorithm was implemented in the MIP package CPLEX version 9. MGA and MCO algorithms were programmed in Microsoft Visual C++ (version 6). All the computational tests were run on an IBM PC (with Intel® 1.73 GHZ, 990 MB RAM). The genetic and random algorithms were allowed to run for number of evaluations = 150 × number of suppliers. The result for one problem instance, with 20 suppliers and 10 items, is shown in Fig. 1.

The essential purpose of developing approximate algorithms is to find solutions as close as possible to the exact solutions. Thus, a crucial measure to evaluate the quality of solutions of approximate algorithms to multiple-objective problems is the distance from the exact solutions. Examining Fig. 1, the solutions found by the genetic algorithm are close to the exact solutions, while the randomly generated solutions are farther away. Sayın (2000) has provided three measures to evaluate the quality of approximate algorithms in Pareto-optimal solutions of multi-objective problems. *Coverage* is a measure of how well the approximate Pareto-optimal solutions represent the total efficient frontier. As can be seen in Fig. 1, the genetic algorithm as well as the Monte Carlo generation are able to provide solutions quite close in coverage to the exact

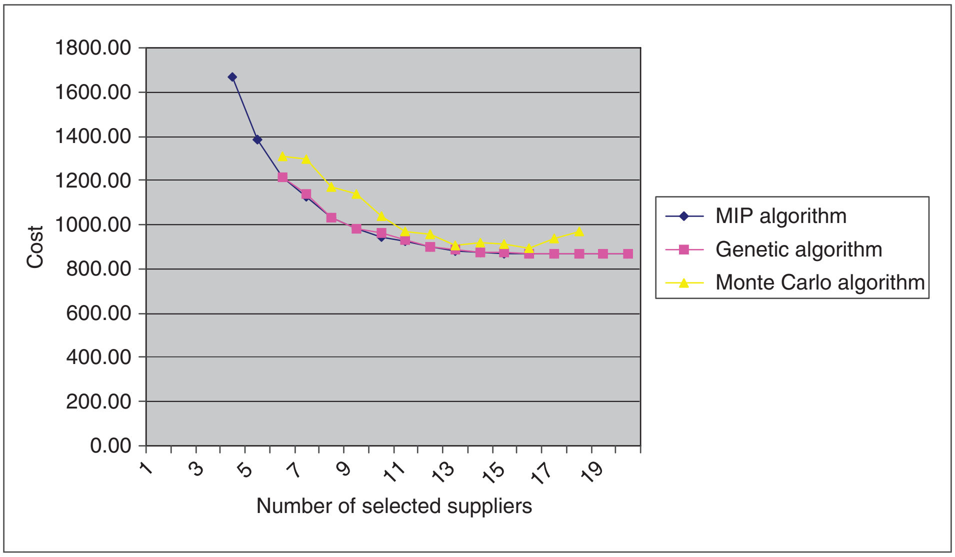

Fig. 1. Pareto-optimal solutions provided by different algorithms.

solutions. *Uniformity* is the ability to find solutions uniformly distributed around the solution space. Points in clusters are not desirable. Again, the genetic and random algorithms are able to capture a good distribution of the solutions. *Cardinality* relates to the number of solutions provided. The number of solutions provided by the genetic algorithm is close to the ones generated by MIP. Monte Carlo generation is slightly worse in this regard.

To further test the algorithms, 10 problems of various sizes (here, problem size refers to the number of suppliers and the number of items in a problem) were generated and solved by the algorithms. The results are shown in Table 1. The cardinality measure is represented in the table by the average number of Pareto-optimal solutions found by the algorithm. The cost of the solution is measured by the average ratio of the cost found by an algorithm to the cost found by the genetic algorithm. The average times taken to generate the Pareto-optimal solution frontiers are also shown in the table. To ensure parity in comparison, the total number of solutions generated by the Monte Carlo algorithm was the same as the number of solutions generated by the genetic algorithm.

The results show that the computational times for the MIP algorithm grow very quickly as the problem size increases. The MIP algorithm could not compute the efficient frontier for problems with 2550 variables (of which 50 were binary), and 5150 constraints within the 2-h time limit. For the problems able to be solved by MIP, the solutions found by the genetic algorithm are close to the solutions found by MIP, the ratio of the costs ranging from 94% to 99%. However, the genetic algorithm is much quicker. The Monte Carlo algorithm and the genetic algorithm take about the same time, as can be expected because the same number of chromosomes are evaluated. MGA provides 4–13% better solutions than the Monte Carlo algorithm for the smaller problems, but for the biggest problems the solution quality of MGA is not much better than that of the

Table 1
Computational results

Problem size	MIP			Genetic Algorithm		Monte Carlo		
	Average number of solutions	Average cost ratio to genetic	Average time in seconds	Average number of solutions	Average time in seconds	Average number of solutions	Average cost ratio to genetic	Average time in seconds
20, 10	16.9	0.9885	1.05	16.2	< 1	12.9	1.093	< 1
20, 20	15.5	0.9936	4.72	14.4	< 1	11.7	1.067	< 1
20, 40	14.3	0.9930	27.03	13.2	< 1	10.6	1.044	< 1
50, 25	46.0	0.9419	511.3	39.2	4.11	26.1	1.126	2.06
50, 50			*	39.0	5.70	25.5	1.103	3.70
50, 100			*	36.8	12.41	25.5	1.070	9.10
100, 50			*	67.7	30.7	37.9	1.045	28.9
100, 100			*	67.9	79.04	38.9	1.041	63.6
100, 200			*	65.8	222.2	35.6	1.035	173.9
200, 100			*	105.3	584.2	56.4	1.006	769.6
200, 200			*	105.8	1546.5	56.3	1.005	1522.9
200, 400			*	103.8	4308.1	56.7	1.005	4447.9

*Efficient frontier could not be computed within 2 h.

Monte Carlo algorithm. However, MGA performs much better than Monte Carlo with regard to the cardinality of the solutions. The performance of MGA in this regard improves in comparison as the problem size increases. For the largest problems, the Monte Carlo algorithm generates only about half the number of solutions generated by MGA.

6. Discussion of the efficient frontier

Figure 2 shows the efficient frontier for different levels of the ratio of suppliers to number of items. The graphic shows the result for 10 replications, with the number of suppliers fixed at 20. Obviously, the total cost increases as the number of items increases. The frontier is also flatter as the number of items decreases. The opportunity for supplier reduction clearly reduces as the number of items increases vis-à-vis the number of suppliers.

The purpose of generating Pareto-optimal solutions is to make the trade-offs clear to the decision maker, so that they can figure out their preferences and make a decision. The cost of supplier rationalisation or vendor reduction programmes can be visualised clearly from the Pareto-optimal frontier. The exact MIP algorithm may be used for smaller problems and the genetic algorithm may be used for larger problems. As an example, the percentage increase in cost for a simulated problem with number of suppliers at 20 and various number of items is shown in Fig. 3.

With the ratio of number of items to number of suppliers at 1:1, the cost does not increase at all until the number of suppliers is reduced by 25%. A 40% reduction in number of suppliers results in only a 4% increase in costs. When the items:suppliers ratio increases to 2:1, 20% reduction in the number of suppliers results in a 4% increase in costs. Thus, with our limited simulated data,

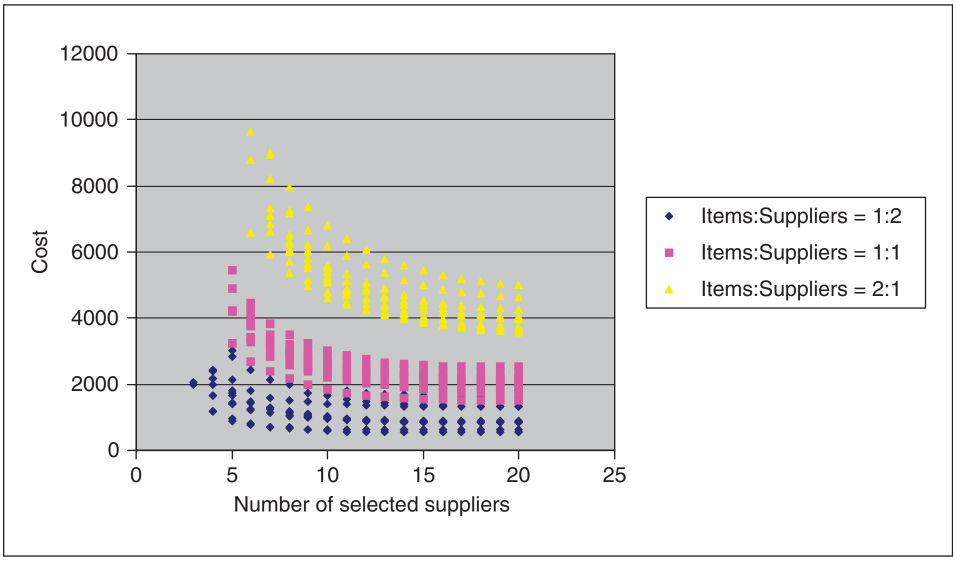

Fig. 2. The efficient frontier for different ratios of items to suppliers.

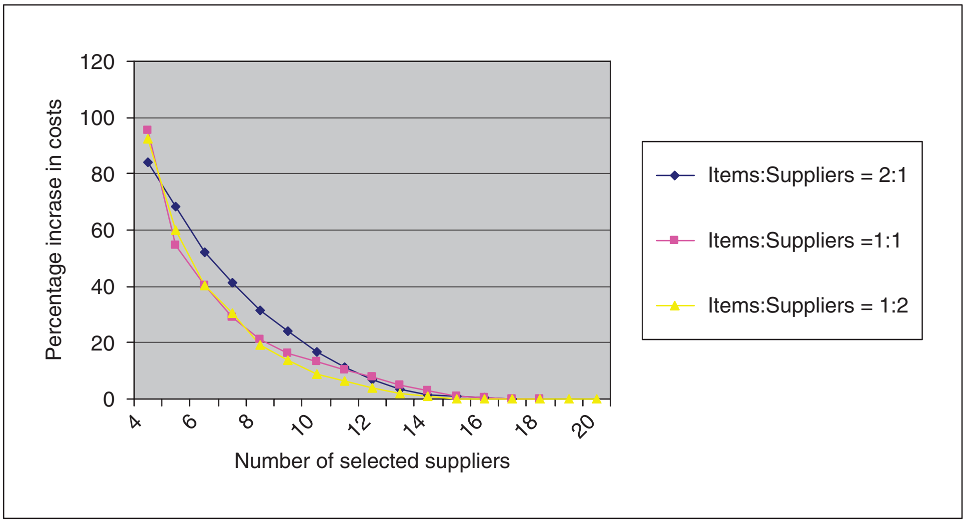

Fig. 3. Increase in costs due to reduction in the number of suppliers.

this suggests that the tail of this curve is quite flat, and the management goal of supplier reduction may be achieved with not very significant increases in costs.

7. Conclusion

The contributions of this paper are twofold: the inclusion of number of selected suppliers as a criterion in supplier selection in an MIP model and the development and testing of a multi-population genetic algorithm for the generation of Pareto-optimal solutions.

We presented a supplier selection problem with two criteria: number of suppliers selected and the cost of purchase. Quality and delivery performance were modelled as constraints. Three algorithms were presented to generate the Pareto-optimal solutions. The exact (MIP) algorithm solved problems with up to 50 suppliers within reasonable time, but not higher-sized problems. The performance of multi-population genetic algorithm was close to the MIP algorithm results. The genetic algorithm performed better than MCO, particularly in regard to the number of solutions generated. In regard to the purchase cost of the solutions, the genetic algorithm performed better for all but the largest problems for which the performances of the two algorithms were almost even.

This paper focused on two objectives: number of selected suppliers and the total cost of purchase. However, the literature shows that the objectives of quality and delivery are important as well. Previous authors (Weber and Current, 1993) have discussed the cost–quality–delivery tradeoffs, but the number of suppliers was not included as an objective. The multiple trade-off between these objectives is worthy of further investigation.

References

Barbarosoglu, G., Yazgac, T., 1997. An application of the analytic hierarchy process to the supplier selection problem. *Production and Inventory Management Journal* 38, 1, 14–21.

Buffa, F.P., Jackson, W.M., 1983. A goal programming model for purchase planning. *Journal of Purchasing and Materials Management* 19, 3, 27–34.

Cochran, J.K., Horng, S.-M., Fowler, J.W., 2003. A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines. *Computers and Operations Research* 30, 7, 1087–1102.

Current, J.R., Weber, C.A., 1994. Application of facility location modeling constructs to vendor selection problems. *European Journal of Operational Research* 76, 3, 387–392.

Dahel, N.E., 2003. Vendor selection and order quantity allocation in volume discount environments. *Supply Chain Management: An International Journal* 8, 4, 335–342.

Degraeve, Z., Roodhooft, F., 1999. Improving the efficiency of the purchasing process using total cost of ownership information: the case of heating electrodes at Cockerill Sambre S.A. *European Journal of Operational Research* 112, 1, 42–53.

Dickson, G.W., 1966. An analysis of vendor selection systems and decisions. *Journal of Purchasing* 2, 1, 5–17.

Ehrgott, M., Gandibleux, X., 2002. Multiobjective combinatorial optimization. In Ehrgott, M., Gandibleux, X. (eds) *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*. Kluwer's International Series in Operations Research and Management Science, No. 52, Kluwer Academic Publishers, Berlin, pp. 369–444.

Ghodsypour, S.H., O'Brien, C., 1998. A decision support system for supplier selection using an integrated analytic hierarchy process and linear programming. *International Journal of Production Economics* 56–57, 199–212.

Jayaraman, V., Srivastava, R., Benton, W.C., 1999. Supplier selection and order quantity allocation: a comprehensive model. *The Journal of Supply Chain Management* 35, 2, 50–58.

Jones, D.F., Mirrazavi, S.K., Tamiz, M., 2002. Multi-objective meta-heuristics: an overview of the current state-of-the-art. *European Journal of Operational Research* 137, 1, 1–9.

Karpak, B., Kumcu, E., Kasuganti, R., 1999. An application of visual interactive goal programming: a case in vendor selection decisions. *Journal of Multi-Criteria Decision Analysis* 8, 93–105.

Min, H., 1994. International supplier selection: a multi-attribute utility approach. *International Journal of Physical Distribution and Logistics Management* 24, 5, 24–33.

Murata, T., Ishibuchi, H., Tanaka, H., 1996. Multi-objective genetic algorithm and its applications to flowshop scheduling. *Computers and Industrial Engineering* 30, 4, 957–968.

Narasimham, R., 1983. An analytical approach to supplier selection. *Journal of Purchasing and Materials Management* 19, 4, 27–32.

Sayın, S., 2000. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming* 87, 3, 543–560.

Weber, C.A., Current, J.R., 1993. A multiobjective approach to vendor selection. *European Journal of Operational Research* 68, 2, 173–184.

Weber, C.A., Ellram, L.M., 1993. Supplier selection using multi-objective programming: a decision support system approach. *International Journal of Physical Distribution and Logistics Management* 23, 2, 3–14.

Weber, C.A., Current, J.R., Benton, W.C., 1991. Vendor selection criteria and methods. *European Journal of Operational Research* 50, 1, 2–18.

Weber, C.A., Current, J.R., Desai, A., 2000. An optimization approach to determining the number of vendors to employ. *Supply Chain Management: An International Journal* 5, 2, 90–98.