J Intell Manuf
DOI 10.1007/s10845-012-0667-0

Solving a multi-objective master planning problem with substitution and a recycling process for a capacitated multi-commodity supply chain network

Ching-Chin Chern · Seak-Tou Lei · Kwei-Long Huang

Received: 20 April 2010 / Accepted: 31 May 2012
© Springer Science+Business Media, LLC 2012

Abstract This study focuses on solving the multi-objective master planning problem for supply chains by considering product structures with multiple final products using substitutions, common components, and recycled components. This study considers five objectives in the planning process: (1) minimizing the delay cost, (2) minimizing the substitution priority, (3) minimizing the recycling penalty, (4) minimizing the substitution cost, and (5) minimizing the cost of production, processing, inventory holding and transportation. This study proposes a heuristic algorithm, called the GA-based Master Planning Algorithm (GAMPA), to solve the supply-chain master planning problem efficiently and effectively. GAMPA first transforms the closed-loop supply chain into an open-loop supply chain that plans and searches the subnetworks for each final product. GAMPA then uses a genetic algorithm to sort and sequence the demands. GAMPA selects the chromosome that generates the best planning result according to the priority of the objectives. GAMPA plans each demand sequentially according to the selected chromosome and a randomly-selected production tree. GAMPA tries different production trees for each demand and selects the best planning result at the end. To show the effectiveness and efficiency of GAMPA, a prototype was constructed and tested using complexity analysis and computational analysis to demonstrate the power of GAMPA.

Keywords Supply chain management · Advanced planning and scheduling · Master planning · Heuristic algorithm · Multiple-goal optimization · Substitutions · Recycle process · Recycling penalty

C.-C. Chern (✉) · S.-T. Lei
Department of Information Management, National Taiwan University, 50, Lane 144, Sec. 4, Keelung Road, Taipei 10625, Taiwan
e-mail: cchern@ntu.edu.tw

S.-T. Lei
e-mail: r95725019@ntu.edu.tw

K.-L. Huang
Institute of Industrial Engineering, National Taiwan University, No. 1, Sec 4, Roosevelt Road, Taipei 106, Taiwan
e-mail: craighuang@ntu.edu.tw

Introduction

In a global competitive environment, a regional crisis or disaster, such as an earthquake and a tsunami, may jeopardize the operations of an entire supply chain. In order to diminish the impact of uncertainty, it is common for a company to maintain a list of alternative components, especially for the critical parts. Seeking for substitute components ensures the order fulfillment process and reduces the possibility of shortage but may incur extra sourcing costs. In this study, the priorities of using substitutes and associated costs are explicitly considered while alternative components are presented in product structures of finished goods. Following the trend of the eco-conscious, more and more governments are urging manufacturers to take responsibility for the recovery of returned products or to reuse returned parts. Because the costs of violating recovery regulations are huge and the benefits obtained from remanufacturing and green reputation are substantial, it is crucial to integrate the recycling processing into the supply chain design and planning. Thus, this study takes into consideration of the penalty on the unfilled recovery rate.

This study provides a method for generating a production plan for multiple final products and a supply chain network that will satisfy all the demands, while minimizing the delay costs, the substitution priorities, the recycling

Published online: 12 June 2012

penalties, the substitution costs and the costs of production, transportation and inventory holding—all while respecting the capacity limitations and the demand deadlines. A supply chain with substitution and a recycling process has many different partners, including materials vendors, manufacturers, distribution centers, retailers, collectors, disassemblers and reprocessors. A supply chain planning problem addresses the difficulties involved in synchronizing the manufacturing processes, transporting materials, semi-finished products, final products, and recycling of used products all along the supply chain. Solving such problems facilitates decision-making related to the effective and efficient use of production and transportation capacities over periods ranging from 1 month to 1 year.

In recent years, these multi-objective problems have attracted the attention of many researchers in different domains for many reasons, such as the nature of the multi-objective problem, the recognized concerns of numerous stakeholders, and the enormous improvements in the computing systems (Demirtas and Ustun 2009; Evans 1984; Pirkul and Jayaraman 1998; Sawik 2007). In the past, the multiple objectives involved in a master planning problem usually included customer satisfaction and inventory holding. In a competitive business environment, customer satisfaction is the top priority when filling the demands for a single entity in a supply chain (Kelbel and Hanzálek 2011; Sawik 2007) or even for an entire supply chain (Chern and Hsieh 2007). In supply chain operations, limited capacities among partners sometimes cause delays, or even cancellations, of demands. These delays can cause lost sales and profits, which implies that delays should be avoided whenever possible. Thus, minimizing the number of tardy orders or delay costs is usually the primary optimal objective (Chern and Hsieh 2007; Sawik 2007). On the other hand, due to capacity limits, holding inventory is sometimes necessary in order to fill the demands before their due date, and the secondary objective in some research is thus to minimize the inventory held (Chern and Hsieh 2007; Demirtas and Ustun 2009; Pirkul and Jayaraman 1998; Sawik 2007). However, some industries produce and deliver to customers directly without holding inventory, (e.g., computer assembling and catering industries) (Sawik 2007).

In addition, due to increasing government involvement (through regulatory measures) and the competitiveness of the supply chain members, the multiple objectives in the planning problems now include minimizing the recycling penalty and minimizing the substitution costs. Unlike the total contradiction between the typical objectives, (i.e., customer satisfaction and inventory holding), these new objectives neither totally contradict each other and nor totally complement each other. Most of the previous solution methods used the contradictory nature of the objectives to solve the planning problems. Adding the new set of objectives into the planning problems makes the solutions obtained by the typical solution method unrealistic or infeasible. Therefore, it is important to find a new solution method for this type of multi-objective master planning problem.

In light of due date constraints and capacity limitations, manufacturers or producers often seek substitutions from outside the original supply chain structure to compensate for shortages of important parts. However, such substitutions do not have unlimited potential. A substitution component is not the original component, and thus using substitute components may lower the product quality or cost more and consequently should be avoided as much as possible. Balakrishnan and Genunes (2000) called the planning problem with substitutions, "Requirements Planning with Substitutions (RPS)", and constructed a search algorithm based on the Shortest Path method and the Dynamic Programming to find the optimal solutions. Genues (2003) proposed a heuristic algorithm for solving the uncapacitated facility location problem (UFLP). Both studies considered only problems that had a two-level Bill of Material (BOM) structure with independent substitutions and without capacity limitations and substitution priorities. This paper will examine planning problems that have BOM structures with more than two levels, with general substitutions and under the capacitated conditions.

With the growing concerns over environmental issues in the past decade and the progress in recycling technology, the recycling of used products is now well developed and the use of recycled items as the raw materials for new products is routine in several industries, such as the pulp and paper industry and the metal industry (Ashley 1993; Schultmann et al. 2003). Although integrating the recycling process into supply chain operations with several members is crucial, previous studies tempt to focus on applications only for a single stakeholder or for a small number of supply chain members (Ashley 1993; Sheu et al. 2005; Stock 1992). Mabini and Gelders (1991) proposed a deterministic optimization model to solve item-repair problems for a multi-item repair system, assuming constant demand and return rates. In his study, Richter (1996a,b) used a deterministic mathematical model to seek the optimal parameters for different inventory control policies and discussed the influence of return rates on the policies. To remedy the limitations of the previous studies, Sheu et al. (2005) proposed a multi-objective mathematical model that considered an integrated green supply chain operation, which combined a product manufacturing supply chain with recycled-product reverse logistics. Our paper will solve planning problems involving supply chain operations with recycling like the one proposed by Sheu et al. (2005) but with five objectives in the planning process.

A multi-objective master planning (MP) problem deals with the various product structures that can result from different final products, seeking to create a feasible plan for the demands of multiple final products in a supply chain in order to achieve certain predefined goals (e.g., minimizing delay penalties or minimizing the costs of production, transportation, substitution and inventory holding). Most of the previous solution methods used the contradictory nature of the objectives to solve multi-objective master planning problems. When adding substitution and a recycling process to these master planning problems, the objectives become even more complex, minimizing the recycling penalties, the substitution costs, the substitution priorities, and the costs of production, transportation and inventory holding. Because the new multi-objective master planning problem lacks the total contradictory nature of the objectives, previous solution methods cannot be used and thus developing a new solution method for this type of multi-objective master planning problem is crucial.

In the solution processes of the many studies, linear programming (LP), nonlinear programming (NLP) or mixed integer programming (MIP) techniques are often adopted, especially in the problems related to Master Planning (MP) (Beamon 1998; Evans 1984; Erenguç et al. 1999; Lakhal et al. 2001; Mabini and Gelders 1991; Moon et al. 2006; Richter 1996a,b; Sheu et al. 2005). However, even if Bender's decomposition, the Primal/Dual Network Simplex Algorithm, and/or other modified optimization methods (Beamon 1998; Lee et al. 2002) are applied to facilitate the solution process, finding the optimal solution for the master planning problem with substitution and a recycling process is still very difficult (Evans 1984). Even the general-cost version of the single final product uncapacitated planning problem has been proven to be NP-hard (Evans 1984; Richter 1996b), so it is no surprise that the more complicated multi-products capacitated problems, which incorporate multi-level supply chain networks, have also been qualified as NP-hard.

Jayaraman and Pirkul (2001), Pirkul and Jayaraman (1998) have used an MIP formulation in an integrated Advance Planning System (APS) model and have developed a heuristic procedure relying on Lagrange Relaxation. However, their algorithm has trouble solving the problem efficiently and effectively as it becomes more complicated. Even when their algorithm does find a solution, the Lagrange Relaxation only guarantees that the solution will be a lower-bound solution, but it may be neither optimal nor feasible.

Chern and Hsieh (2007) introduced a heuristic Multi-Objective Master Planning Algorithm (MOMPA) that considers more than one objective and includes both inventory holding costs and delay penalties, as well as common components. Chern and Yang (2006) further focused on solving the master planning problem for supply chains by considering substitutions and common components. In addition, Chern and Huang (2007) solved the master planning problem for supply chains by considering a recycling process. Nonetheless, substitutions and recycling processes are not simultaneously dealt with in any of the above studies.

Methods based on Genetic Algorithms (GA) have been widely used to solve the master planning problem. GA are implemented in a computer simulation in which a population of abstract representations (called chromosomes) of possible solutions to an optimization problem evolves toward better solutions (Awadh et al. 1995; Lee et al. 2002; Moon et al. 2002, 2006; Okamoto et al. 2006; Vergara et al. 2002). Genetic algorithms are a particular class of evolutionary algorithms (EA) that use techniques inspired by evolutionary biology, such as inheritance, mutation, selection and crossover (Awadh et al. 1995; Lee et al. 2002; Moon et al. 2002; Vergara et al. 2002). GA-based methods have been used in the studies of Lee et al. (2002), Lee and Kim (2002), Awadh et al. (1995), Moon et al. (2002, 2006) and Vergara et al. (2002) to solve various types of planning problems in supply chain networks. However, these studies did not address the problems faced by the GA, namely the problems due to the mechanisms for finding the initial solution, the fitness function, the crossover rule, the mutation process, and the stopping condition. The inefficiency of the GA-based methods in the event of increasing numbers of customer demands has also been ignored in the literature (Brailsford et al. 1999).

As mentioned before, the MIP model is not applicable for large-scale multi-objective master planning problems. Even if the MIP model is solvable, the optimal solution may not lead to a tighter lower bound due to the multi-stage optimization process for the multiple sequential objective functions or to poor weight assignments for the weighted-sum objective functions. Using the multi-stage optimization process, the first objective function is minimized, then the second, and so on; using the weighted-sum optimization process, weights have to be determined first.

In reality, management not only wants to avoid delays by using a prioritized substitute but also wants to use substitutes whenever the total cost can be minimized. Therefore, the multi-objective functions are considered simultaneously with different accounts (and shifting weights). The MIP model does not have enough flexibility to switch the order of the functions or change the weights of these functions in solution process. In the MIP model, the order (or weights) of these objective functions is fixed in each problem. Therefore, MIP may obtain an optimal solution that is not exactly what the management had in mind.

Heuristic approaches, on the other hand, have the flexibility to take these objective functions into account by evaluating the delay penalty, the substitution priority, the recycling penalty, the substitution cost and the inventory holding cost to determine whether or not switching to the substitute items or holding inventory is necessary to avoid delay penalties. As a result, a heuristic approach may produce a solution that is not optimal (compared to the solution from the MIP model) but is acceptable to the management.

Most of the previous heuristic approaches used the contradictory nature of the objectives (e.g., delay penalty and inventory holding cost) to solve multi-objective master planning problems. However, when considering more objectives, the heuristic approaches can no longer consider only the contradictory nature of the objectives but also need to simultaneously weight all the objectives, which is exactly the strength of a GA-based method. This study will propose a heuristic algorithm embedded with a Genetic Algorithm, which thus produces better solutions that are acceptable to the management.

The rest of the paper is organized as follows. Section "Problem description and multi-objective functions" describes the problem. Third and fourth section presents, respectively, our GA-based Master Planning Algorithm (GAMPA) and our Heuristic Planning and Fitness Evaluation Algorithm (HPFEA). Section "The GAMPA solution process and complexity analysis" demonstrates the solution process of the heuristic algorithm using a simple example. Section "Computational analysis" compares the results obtained with our heuristic algorithm to those obtained with the MIP method, in order to evaluate the algorithm's efficiency and optimality. Finally, section "Conclusions" offers our conclusions and suggestions for future research.

Problem description and multi-objective functions

In the process of filling demands, the supply chain partners take responsibility for different tasks. Each partner generates its own costs (e.g., production, transportation, or inventory) according to the tasks performed. It is in the best interest of the whole supply chain to fill all demands, allotting the resources in the most efficient way, thus minimizing the supply chain's overall costs.

In a competitive business environment, customer satisfaction is the top priority when filling the demands for the whole supply chain. Delays can cause lost sales and profits, which implies that delays should be avoided whenever possible. However, sometimes delay is inevitable due to capacity limitations and demand due dates. In these circumstances, it is important to allocate the production resources to the demands according to the importance of the customer (i.e., the size of the delay penalty).

In order to respond quickly to market conditions and avoid delaying promised demands, substitutions from outside the regular supply chain may be necessary and should be included in the planning process. However, substitutions do not have unlimited potential and are not the original components. Therefore, using substitutions may lower the product quality or cost more and consequently, should be avoided as much as possible. Sometimes, using substitutions requires modifying the regular production process, which results in longer processing times and additional fixed and variable costs.

The Bill of Material (BOM) considered in this paper contains several final products, as well as the various components and the items that can be substituted for them. Lyon et al. (2001) divided substitutions into two categories: alternative BOM and general substitution. In the former, different production processes must be adopted in order to use the original materials and their substitutions, to complete the semi-finished products or final products. In the latter, however, no change is necessary; the same production processes can be used. Given that alternative BOM can be converted to general substitution by using virtual items, as proposed by Chern and Yang (2006), this study assumes that all substitutions in BOM structures are general substitutions. Balakrishnan and Genunes (2000) have identified two types of general substitutions: independent and interacting. Interacting substitution can be transformed into independent substitution, by using virtual items, as proposed by Chern and Yang (2006). Therefore, this study includes general independent substitutions in the planning problem.

When integrating a recycling process into a supply chain operation, the corresponding supply chain structure is changed from an open-loop to a closed-loop, and the product structure is changed from a tree to a loop. A supply chain network with a recycling process can be divided into two parts: the product manufacturing supply chain and recycled-product reverse supply chain. In addition, a product structure with recycled parts can also be divided into two BOMs: one for manufacturing and one for recycling. Because a recycled component may be substituted at the manufacturing node, virtual items should be used to prevent wrong items being used in future recycling and to solve the closed-loop problem.

An example of the supply chain and product structure is shown in Figs. 1 and 2. Figure 1 shows two product structures for final products, P_5 and P_6, with manufacturing and recycling BOMs. P_5 is recycled at nodes D_1, D_2, and D_3 in order to obtain recycled parts, P_1 and P_4, while P_6 is recycled at node D_4 in order to obtain recycled part P_1. This situation is common in the auto manufacturing industry: a spark plug can be recycled either from an oil change procedure or from a junk car. The similar situation also occurs in the recycling processes for batteries or printer cartridges. Figure 2 shows a closed-loop supply chain network for the two final products, P_5 and P_6, with two stages, one before the virtual items are added and the other after.

The recycling penalty has been a key factor in motivating manufacturers to use recycled parts. This penalty helps to "force" manufacturers to reuse parts when producing their products. Target recycling ratios are set for different products and are applied to the main recycled components of each final sellable product. Manufacturers of a specific final product must use enough recycled parts to meet the target ratio

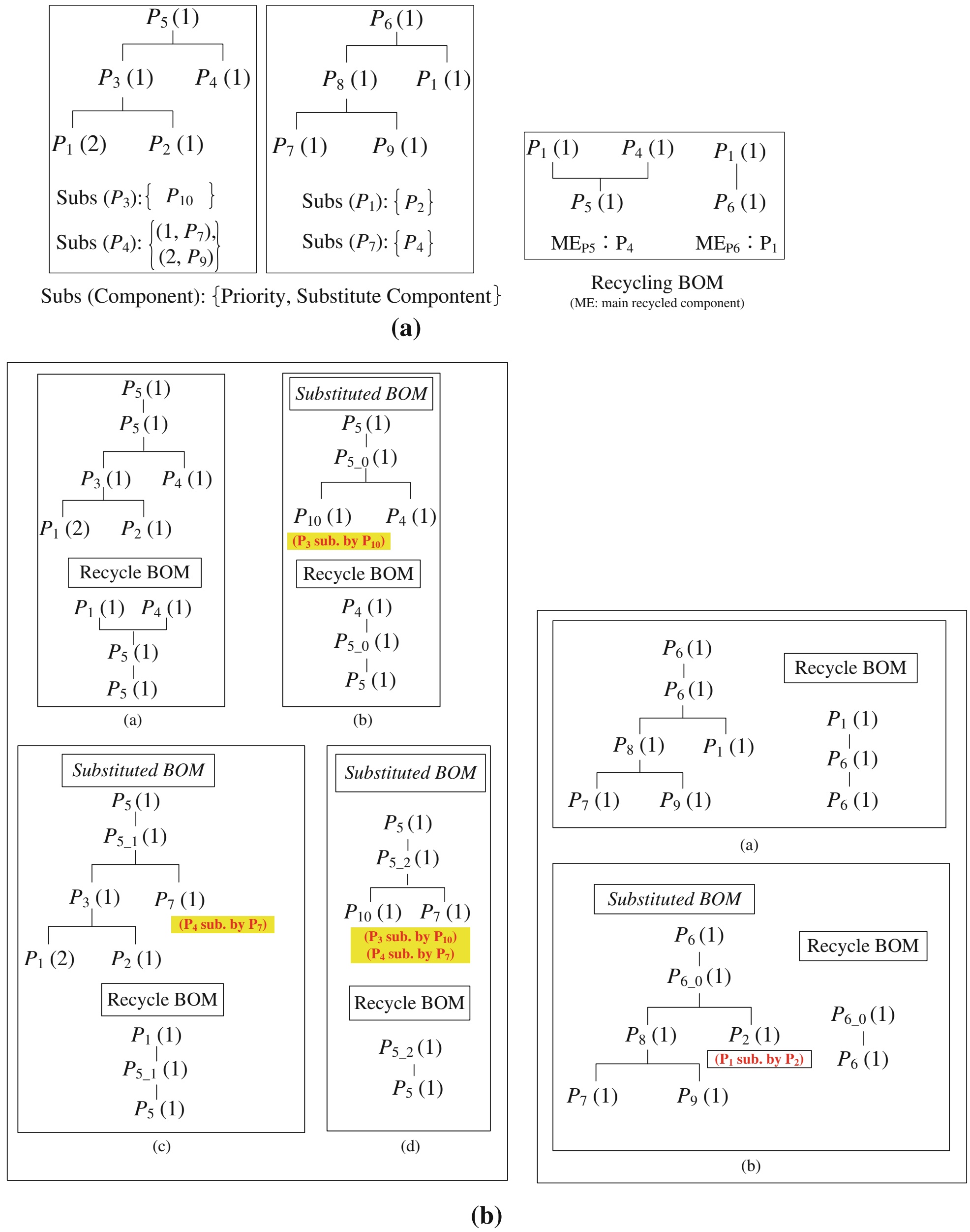

Fig. 1 An example of a BOM structure. **a** Before conversion, **b** after conversion

and thus avoid the recycling penalties. For example, in Fig. 2, if the target ratio for P_5 at manufacturing node M_2 is set at 60 % and the demand quantity is 100, then to avoid recycling penalty, 60 units of P_5 at node M_2 must be made using recycled part P_4. In this study, target recycling ratio is assumed to be predetermined and constant in planning problems.

Recycling penalties can be fixed or variable. This means that a manufacturer will pay either a set fee if it doesn't use enough recycled products and/or a fee that varies depending on the number of units under the target recycling ratio. When the target recycling ratio is not met, a fixed recycling penalty is applied to the manufacturer once each period. The variable recycling penalty is calculated as the variable recycling penalty per unit multiplied by the number of units under the target ratio. This number is calculated as the demand quantity multiplied by target recycling ratio minus the actual number of recycled units used.

Another major issue when solving planning problems with recycling process is predicting the recycling pattern. As Fleischmann et al. (2002, 2000) pointed out, if a constant return rate is assumed for a deterministic model, the search for solutions may be limited to finding the optimal solution in a given operational environment. However, in a real green supply chain operation, the recycling pattern is uncertain and

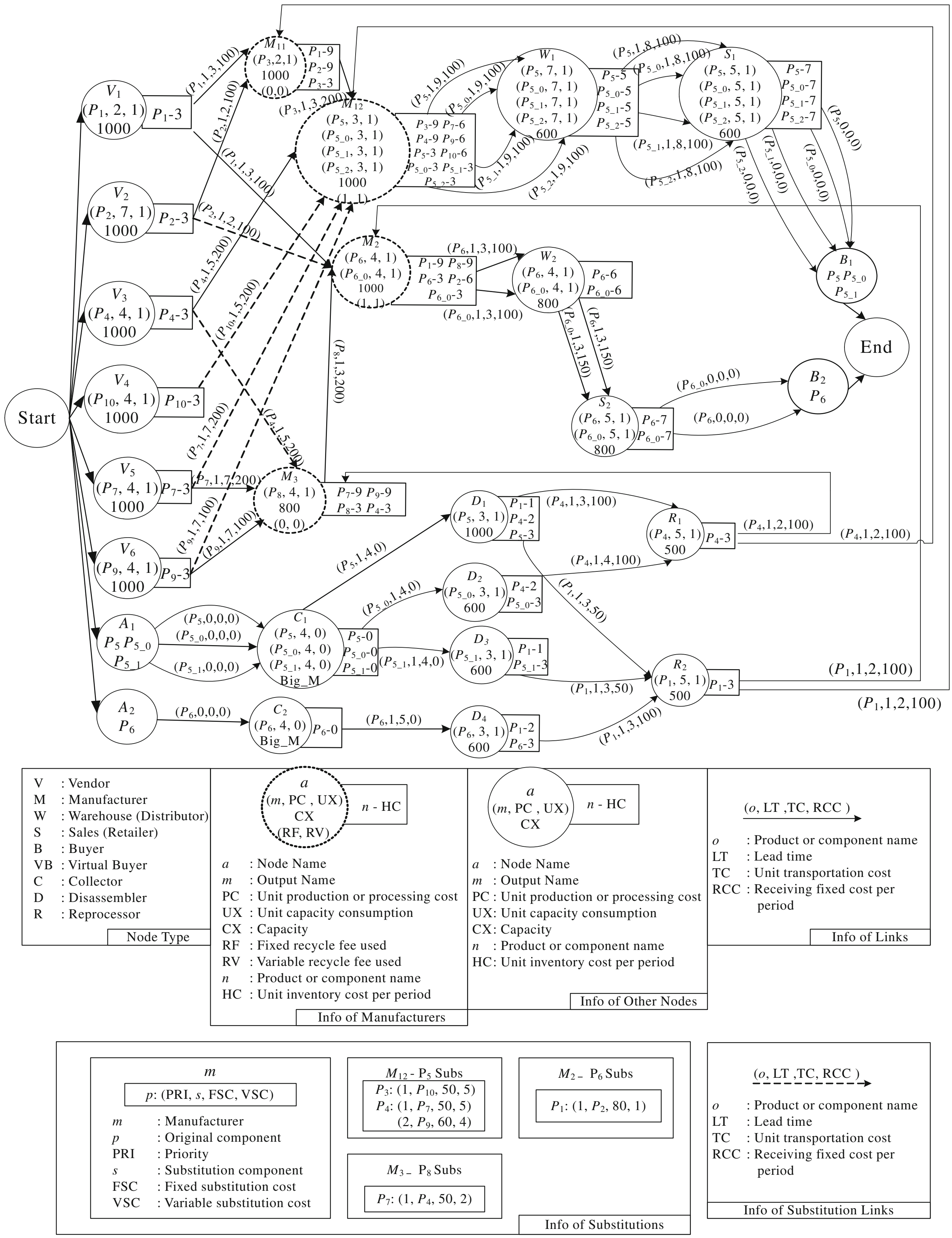

Fig. 2 An example of a supply chain network

not constant. As a response to this uncertainty, in this study, we assume a stochastic recycling process and thus use a mathematical optimization model based on the long-term average of this recycling process.

We used a multinomial distribution to model the stochastic recycling pattern. Multinomial distribution has several parameters: the total number of outcomes and the success probability for each outcome. We assume that, once a product is sold, its recycling time will follow a multinomial distribution, with the total number of recycling periods equal to T. The probability that the final product p will be recycled in period t for component m is $PROB_{pmt}$ where $\sum_{t=1}^{T} PROB_{pmt} = 1$. Under this assumption, the average number of recycled products used in each period follows the sales in previous periods. Thus, this number is known and constant, and can be used to formulate a mixed integer programming (MIP) model and the heuristic planning algorithm.

The objective of this study is to provide a method for generating an optimal production plan that will satisfy all demands, while minimizing delay penalties, substitution priorities, substitution costs, recycling penalties, and the costs of production, transportation and inventory holding—all while respecting the capacity limitations, substitution limitations, recycling requirements and demand deadlines. Unlike the contradictory nature of delay penalties and inventory holding costs, these objectives do not totally conflict with each other nor are they completely complementary. In addition, the solution process needs to be flexible in order to be able to change the order of these objectives depending on the manufacturer's priorities. Since none of the existing solution methods is able to solve such complicated problems, it is important to find a new solution method for this type of multi-objective master planning problem.

Apart from the delay penalties, substitution priorities, the recycling penalties, and the recycling pattern, the following information about products, facility capacities and demand requirements are also needed to understand how the MIP model is constructed and how the heuristic algorithm functions.

(1) It is assumed that the planning time horizon is divided into intervals, called "time buckets." Furthermore, it is also assumed that each node receives the products from its upstream nodes at the beginning of a time bucket, with the products being ordered a few time buckets ago depending on the shipping lead time from that upstream node to this node; each node delivers the products to the downstream node at the end of a time bucket, at which point the inventory level and the inventory holding cost are determined.

(2) It is assumed that more than one final product is produced or transferred in a given supply chain network. The structure of these final products is presented in the form of product tree graphs, whose nodes represent items and whose links represent parent-child relationships (Fig. 2). In the figure, the numbers in the parentheses beside the item names show the quantities of child items needed to make one final parent item. The BOM also provides substitutions information. As mentioned above, this study assumes that all substitutions in BOM structures are general substitutions and that once a symmetrical substitution has taken place, no substitution involving the same items can occur in the opposite direction later on. The final product, P_5, is produced from one P_4 and one P_3, while a P_3 is made from two P_1 and one P_2 in the manufacturing BOM (Fig. 1). In the recycling BOM, one P_1 and one P_4 can be recovered from one P_5, which then can be reused to create a P_5. Therefore, P_1 and P_4 are both parent items and child items, depending on the direction in which the graph is read. P_4 is the main recycled component of the final sellable product P_5, used to set the target recycling ratio.

(3) A directed graph, $G(V, L)$, is used to represent a supply chain network. The nodes in $G(V, L)$ stand for facilities that produce, process, store, sell, collect, disassemble and reprocess products, and the links stand for the logistical connections between two nodes. Seven types of information accompany each node: node name, total capacity (in units), item name, unit production cost, capacity used per item unit, inventory holding item name and unit holding cost per time bucket. Three types of information are provided for each link: item name, transportation lead time (in time buckets) and unit transportation cost. Three types of substitution information are given for each manufacturing node: substitute component, fixed substitution cost, and unit replacement cost. The unit inventory cost for the nodes other than the manufacturing node is defined in the box besides each node; for manufacturing nodes, the unit inventory costs are defined in the boxes below the supply chain network with their substitution costs. Figure 2 presents an example in which M_{11} (a manufacturing node) can produce 1,000 units of P_3 per time bucket. The unit production cost and the unit inventory holding cost per time bucket for producing P_3 on M_{11} are \$2 and \$3, respectively. It takes \$3 to ship a unit of P_3 from M_{11} to M_{12}, and the transportation lead time bucket is 1. The unit inventory holding cost of P_1 at M_{11} is \$9 per period. For P_7 to replace P_4 at M_{12}, the fixed substitution cost is \$50, and unit replacement cost is \$5; for P_9 to replace P_4 at M_{12}, the fixed substitution cost is \$60 and unit replacement cost is \$4.

(4) Two types of recycling information are given for the manufacturing nodes in a supply chain network: fixed recycling fee tag and unit recycling fee tag. If the tag

is equal to 1 on this particular manufacturing node, the fixed recycling fee mechanism is applied to this manufacturer. The same goes for the variable recycling fee mechanism. Figure 1 presents an example of a supply chain network with a recycling process in which M_{12} and M_2 are final product manufacturers and the fixed and variable recycling fee mechanisms are applicable. A fixed recycling fee of \$300,000 is applied to nodes M_{12} and M_2 when the target recycling ratio (60 % for both P_5 and P_6) is not met. A variable recycling penalty is calculated as \$100 multiplied by the number of units under the target recycling ratio at nodes M_{12} and M_2.

(5) Accompanying each demand are the items, quantities, deadlines, and delay penalties per unit time bucket. It is assumed that the demands are divisible, which means that each demand can be filled by several different combinations of vendors, manufacturers, wholesalers, retailers, collectors, disassemblers, and reprocessors. For example, using the product structure and the supply chain network given in Figs. 1 and 2, a demand of 100 units of P_5 can be filled by a combination of 2 production networks: one production network, $\{V_1, V_2, V_3, M_{11}, M_{12}, W_1, S_1, B_1, C_1, D_1, R_1\}$, producing 70 units, and another production network, $\{R_2, V_2, R_1, M_{11}, M_{12}, W_1, S_1, B_1, C_1, D_1, R_1\}$, producing 30 units.

Though the problem can be formulated as a Mixed Integer Programming (MIP) model as shown in Appendix A, difficulties arise when trying to solve this model due to the number of constraints and variables. This study thus proposes a heuristic algorithm, called GA-based Master Planning Algorithm (GAMPA), to efficiently and effectively solve the planning problem for an integrated supply chain with substitution and a recycling process.

GA-based Master Planning Algorithm (GAMPA)

Planning for an integrated supply chain network with substitution and a recycling process implies multiple objectives: (1) minimizing the total delay cost; (2) minimizing substitutions while respecting substitution priorities; (3) minimizing the substitution cost; (4) minimizing the recycling penalty; and (5) minimizing the total cost of production, transportation and inventory holding. In addition, the solution process must be flexible in order to change the order of these objectives depending on the manufacturer's priorities. Sometimes, management wants not only to avoid delays by using substitution or holding inventory but also to minimize the total cost. Unlike the heuristic master planning (MP) algorithm proposed in this study, MIP models are unable to satisfy these requirements.

The heuristic master planning (MP) algorithm that we propose uses a Genetic Algorithm (GA) to generate the demand sequence and to choose the production network from the supply chain network. Before creating the plan, this algorithm compares the different possible combinations of demand sequences and production network choices to determine the best plan. Whenever the order of the objective functions is changed, the criteria used to select the best plan can be changed accordingly. This heuristic algorithm is called the GA-based Master Planning Algorithm (GAMPA).

The difficulties of solving a planning problem for supply chains with multiple objectives lie in determining which combinations of demand sequences and production network choices is the best for a given order of multiple objective functions. For the example shown in Figs. 1 and 2, many possible production networks can be used to produce the final product P_5. Among them are $\{V_1, V_2, V_3, M_{11}, M_{12}, W_1, S_1, B_1, C_1, D_1, R_1\}$ and $\{R_2, V_2, R_1, M_{11}, M_{12}, W_1, S_1, B_1, C_1, D_1, R_1\}$. If the current demand of P_5 does not use enough recycled components P_4 from S_1, the fixed and variable recycling penalties will be imposed on M_{12}. Three choices are possible to avoid these recycling penalty: produce this demand of P_5 in the previous periods to avoid recycling penalty but carry extra P_5 inventory on M_{12}; produce future demands of P_5 as well as the current demand in this period and carry extra inventory to fill these future demands; or create virtual demands, not real sales demands, simply to have enough recycling volume. We assume that virtual demands are a feasible planning strategy to counteract recycling penalties in the manufacturer's off-peak season when the real demand does not generate enough recycling volume.

Of the five objectives that must be accomplished, the first objective function is minimizing the total delay cost. The remaining four objective functions are optimized based on individual management considerations. Because of the greedy nature of GAMPA, the planning results are significantly affected by the order of demands. When sorting demands for different final products, the possibility of using limited resources, substitutions and/or recycled components to fill different demands on the manufacturing node, thus avoiding the delay penalties, substitution costs and/or recycling penalties, must be taken into consideration. The GAMPA mechanism described in the following paragraphs is used to sort demands and find the best solution.

In general, an Genetic Algorithm consists of four steps: initialization, selection, reproduction, and termination. Once the genetic representation and the fitness function have been defined, the GA initializes a solution population. During each successive generation, a proportion of the existing population is selected to produce a new generation based on the defined fitness function. Then, the Genetic Algorithm improves the population by repeatedly applying mutation, crossover, inversion and selection operators. This generational process is repeated until a stopping condition has been reached.

To use an evolutionary operator to sort and plan demands, first a genetic representation (i.e., a chromosome) must be designed. In GAMPA, this chromosome is a vector used to encode the demand sequence. The size of the vector is equal to the total number of demands. Each element of the vector represents one demand number. For instance, a vector with four elements is used to represent the sequence if there are a total of four demands waiting to be planned. A vector (1, 2, 3, 4) implies that demand 1 is planned first, followed by demands 2, 3 and 4; a vector (2, 4, 3, 1) implies that demand 2 is planned first, followed by demands 4, 3 and 1.

Usually, evolutionary operators randomly generate many individual solutions in order to form the initial population. However, GAMPA only generates two initial solutions: one is a randomly-generated solution and the other is a "seeded" solution that is "planted" in area where optimal solutions are likely to be found using a rule-based procedure. The rule-based procedure used to seed the initial solution is as follows: (1) sort the demands according to their due time buckets in ascending order; (2) sort demands that have the same due time bucket according to their delay penalties in descending order; and (3) sort demands that have the same rank after applying the previous two rules according to their demand quantity in descending order.

In GAMPA, the fitness function was designed based on the five objectives that must be accomplished. First in the sequence is the first objective function (minimizing the total delay cost) and then come the rest of the objective functions in the order desired by the individual planner. During each successive generation, the two solutions in the existing population that are deemed the most fit (as measured by the fitness function described above) are selected to breed a new generation of solutions. The two solutions selected are called "parent" solutions, and the results of their "breeding" are called "child" solutions. After generating two initial solutions and selecting them as "parent" solutions, the next step is to produce the next generation of solutions from those selected by two genetic operators: crossover and/or mutation.

In GAMPA, a single crossover point on both parent vectors is selected randomly. All data beyond that point in both vectors is exchanged between the two parent vectors. The resulting vectors, or chromosomes, are the children. Since each chromosome contains an ordered list of elements, a direct swap would introduce duplicates and ignore necessary candidates in the list. Instead of a direct swap, the chromosome up to the crossover point is retained for each parent. After the crossover point, however, the chromosome is ordered in the order of the opposite parent, obviously skipping the elements that are already in the child's list. For example, if the two parents are (1, 2, 3, 4, 5, 6, 7, 8, 9) and (9, 7, 1, 8, 6, 4, 2, 5, 3) and the crossover point is after the 4th element, then the resulting children would be (1, 2, 3, 4, 9, 7, 8, 6, 5) and (9, 7, 1, 8, 2, 3, 4, 5, 6).

In the evolutionary operator, the purpose of mutation is to allow the algorithm to avoid local optima by preventing the chromosomes in the population from becoming too similar to each other, thus slowing or even stopping evolution. In GAMPA, a mutation operator involves a probability that a genetic sequence will mutate from its original state at random. Every time a mutation operator is triggered, a solution (i.e., a chromosome) is randomly generated as a "child" solution. This mutation process also helps GAMPA to avoid taking only the fittest chromosomes in the population for producing the next generation. Instead, GAMPA takes a random selection with a weighting system that favors better fitted chromosomes.

Using crossover and/or mutation operators produces a "child" solution that usually shares many of the characteristics of its "parents". Each "child" solution has a new pair of parents, and the process continues until a new solution population of an appropriate size has been produced or until a stopping condition has been reached. In GAMPA, before the process is terminated, one of two conditions is verified: (1) a fixed number of generations, N, is reached or (2) the fitness of the highest ranking solution has reached a plateau such that successive iterations (W) no longer produce better results.

The three steps in GAMPA are listed below:

- (P1) Initialize the supply chain network and generate the initial population using the GAMPA mechanism described above (i.e., one randomly-generated solution and one "seeded" solution that is "planted" in area where optimal solutions are likely to be found using a rule-based procedure).
- (P2) Evaluate the fitness of each individual in the population using the Heuristic Planning and Fitness Evaluation Algorithm (HPFEA) described in the next section and rank them according to their fitness.
- (P3) Repeat the following steps until the process is terminated: (stopping condition: number of generations $= N$ or number of successive iterations $= W$)
  - (P3-1) Select two best-ranking individuals to reproduce.
  - (P3-2) Produce a new generation of offspring through crossover and/or mutation.
  - (P3-3) Evaluate the individual fitness of the offspring using the Heuristic Planning and Fitness Evaluating Algorithm (HPFEA) described in the next section.
  - (P3-4) Replace the worst-ranked individuals in the population with these offspring.

In the first step (P1), GAMPA generates the initial solutions (i.e., one randomly-generated solution and one rule-based solution) and forms the initial population using these two solutions. In (P2), GAMPA evaluates the fitness of these solutions using our Heuristic Planning and Fitness Evaluation Algorithm (HPFEA) described in the next section. In (P3), GAMPA verifies the stopping condition and retrieves next two best-ranking individuals to reproduce (P3-1). GAMPA then produces new generation of offspring using a crossover or mutation operator (P3-2). GAMPA again evaluates the fitness of these offspring using our Heuristic Planning and Fitness Evaluation Algorithm (HPFEA) (P3-3). GAMPA replaces the worst-ranked individuals in the initial population with these offspring (P3-4) and goes back to verify the stopping condition, and if necessary, repeats steps (P3-1)–(P3-4).

Like other artificial genetic algorithms, the repeated evaluation of the fitness function for complex problems is the most prohibitive and limiting segment of GAMPA. Finding an optimal solution to complex high-dimensional problems often requires very complicated fitness function evaluations, even using genetic algorithms. For real-world master planning problems, one single fitness function evaluation may require from several hours to several days. In this case, it is necessary for GAMPA to forgo an exact evaluation and to use an heuristic fitness evaluation algorithm that is computationally efficient. This algorithm is explained in detail in the next section.

The Heuristic Planning and Fitness Evaluation Algorithm (HPFEA)

There may be more than one production network in a given supply chain that can be used to fill a demand. For example, Fig. 1 shows the many possible production trees for final product PO5. Among the possibilities are $\{V_1, V_2, V_3, M_{11}, M_{12}, W_1, S_1, B_1, C_1, D_1, R_1\}$ and $\{R_2, V_2, R_1, M_{11}, M_{12}, W_1, S_1, B_1, C_1, D_1, R_1\}$. Given the five objectives that must be accomplished, each production tree provides different costs and capacity conditions to fill a demand. It is necessary to find a production network, S, with enough capacity to fill the demands sorted in steps (P1) and (P3-2). HPFEA chooses the production network randomly and then adjusts the production plan to avoid delay penalties, use substitutions according to their priority, minimize substitution cost, avoid recycling penalties, and minimize the total processing cost.

To facilitate the explanation of HPFEA, two important terms related to production tree selection need to be defined first.

- The production network, S, is the network selected randomly from a given supply chain. Please note that S includes both parts of the supply chain: the product manufacturing supply chain and the recycled product reverse supply chain.
- The **A**vailable **C**apacity of S, AC_S, is computed from the minimum available capacity at each node of S, based on DUE_{rp}, the due date time bucket of final product p for demand r.

The following five steps for HPFEA incorporate the above definitions into GAMPA:

(PL1) Convert the given closed-loop $G(V, L)$ to a open-loop supply chain.

(PL2) If there are no unplanned demands, stop and go to (PL5). Otherwise, retrieve the next demand for final product p of demand r based on the sequences determined in (P1) or (P3-2) and initialize the planning results.

(PL3) Randomly select a production network, S, and calculate AC_S, the available capacity of S, using the HPFEA planning mechanism explained later in this section. Repeat this step K times to obtain a better solution.

(PL4) If $DEM_{rp} \geq AC_S$, allocate the AC_S of all the nodes on S to this demand, where DEM_{rp} is the demand quantity for final product p of demand r. Update $DEM_{rp} = DEM_{rp} - AC_S$; re-set DUE_{rp} to the original due date time bucket; and go to (PL3). Otherwise, allocate the DEM_{rp} of all the nodes on tree S to this demand, and go to (PL2).

(PL5) Compute the delay penalty, the substitution priority, the substitution cost, the recycling penalty, and the total processing cost for this plan.

To minimize substitutions and insure that the correct substitution priorities are followed, HPFEA keeps track of each link's substitution condition, inventory holding period and/or delay condition and keeps adjusting the appropriate costs of substitution, inventory holding and/or delays for each link. When planning a demand with insufficient available capacity, HPFEA allows the nodes with substitutions to be added into the supply chain before the production tree is selected randomly. When the selected production tree is unable to fill a demand, HPFEA will switch to the substitute item that is at the top of the substitution list. If two or more items needing substitutes are located on different BOM levels, HPFEA will choose the one on the highest level first. When both the parent and child items are unavailable, HPFEA replaces the parent item so that the child item will no longer create a bottleneck. Sometimes, the bottleneck item has no substitute, in which case HPFEA searches for the parent items that have substitutes and replaces them.

Finally, to minimize the recycling penalty, HPFEA keeps track of the manufacturing node's recycling condition, inventory holding period and/or delay condition and keeps adjusting the appropriate recycling penalty for the manufacturing node. In order to execute HPFEA, the following threshold variable is computed in step (PL3):

- $Period_{int}$ is the threshold of the inventory holding periods that allow the manufacturer to avoid the recycling penalty at the main manufacturing node i for item n at time bucket t, if n is produced on manufacturing node i and is calculated as $(RFC_{int}/RQTY_{int} + RVC_{in})/HC_{in}$, where RFC_{int} is the fixed recycling penalty at node i for item n at time bucket t; $RQTY_{int}$ is the required recycled component quantity of final product n at node i at time bucket t; RVC_{in} is the variable recycling penalty at node i for item n; and HC_{in} is the unit holding cost of item n at node i.

For each demand, if the selected production network, S, includes recycling nodes and recycled parts, HPFEA first adds the recycling penalty (both fixed and variable) to the main manufacturing node i and then activates the plan-ahead mechanism to find enough recycling volume to fill future demands. Only the demands with the due time buckets falling in the range of the due time bucket of the current demand plus $Period_{int}$ should be considered. This implies that these demands may be able to avoid the recycling penalty without incurring larger inventory holding costs. However, due to the capacity limits and target recycling ratio, not all the demands falling into this range are considered. HPFEA selects and pre-plans these demands based on the sequence determined in (P1) and (P3-2).

For example, when P_5 is produced on node M_{12} for a given demand for 400 units, the \$10,000 fixed recycling penalty is incurred only once if the use of recycled component P_4 does not meet the target, as shown in Figs. 1 and 2. The recycling target ratio, 40 %, of P_5 (converted to 160 units of recycled component P_4) is imposed at node M_{12} at time bucket 8 (Fig. 1). If the variable recycling penalty at node M_{12} for P_5 is \$5 and the unit holding cost of P_5 is \$2, $Period_{int} = (10,000/(40\,\% \times 400) + 5)/2 = 33.75$, which implies that at least thirty inventory holding periods for P_5 using recycled component P_4 at M_{12} can be used to avoid the recycling penalty. If the current demand quantity is only 100 units, the recycling penalty is inevitable. Therefore, HPFEA searches the future demands whose due dates fall within the range of 30 time buckets (from 8 to 38) and pre-plans these demands to have enough capacity. Supposing that a demand with a quantity of 200 at due time bucket 9 is found. HPFEA will pre-plan the entire demand.

For the case in which no future demand falls within the feasible range, HPFEA creates enough "virtual demands" to generate enough recycling volume (Richter 1996b). A "virtual demand", by definition, is a demand that is not needed in the future but is used nonetheless to avoid the recycling penalty because this penalty is too large to ignore. For the above example, if no future demand is found, HPFEA will generate a virtual demand with a quantity of 60 units for M_{12} in order to use recycled component P_4 to avoid recycling penalty.

The stochastic recycling pattern, assumed to follow a multinomial distribution, is used to calculate the recycling quantity. Once the demand is sold to a buyer, the recycling life time is started. The expected recycling quantity is calculated as the demand quantity, both regular and virtual, multiplied by the probability of the assumed multinomial distribution. For example, the multinomial distribution of the P_5 recycling pattern at S_1 is $PROB_{pmt} = \{1, 0.2; 2, 0.3; 3, 0.3; 4, 0.1; \text{scrap}, 0.1\}$, which means that, the recycling probability is 0.2, after one period of P_5 sales, 0.3 after 2 periods, 0.3 after 3 periods, 0.1 after 4 periods, and the probability of being eventually scrapped is 0.1 (Figs. 1, 2). If a demand of 400 units is sold at due time bucket 8, then 80 units will be recycled at time bucket 9, 120 units at time bucket 10, 120 units at time bucket 11, 40 units at time bucket 12, and 40 units will never be recycled. HPFEA can create virtual demands while planning and assign virtual due time buckets to these demands, which are then used to calculate the recycling quantities.

In the following section, a simple MP problem is used to demonstrate GAMPA's solution process and the result obtained with GAMPA is compared to the result obtained using an MIP model. (Please refer to Appendix A for the multi-phase MIP model.)

Table 1 Demand information for the GAMPA solution process

Demand ID	Demand	Due date	Delay cost	Final product	Demand sequence
1	360	8	3	P_5	1
2	360	10	3	P_5	2

The GAMPA solution process and complexity analysis

To demonstrate the GAMPA solution process, a simple MP problem, based on the product structure and the supply chain network shown in Figs. 1 and 2, is solved below. Table 1 provides the demand information for this MP problem.

When solving this problem, GAMPA first initialized the supply chain network graph in step (P1) and then chose two initial solutions: one was chosen randomly and the other using a rule-based sorting mechanism. The results of step (P1) are also given in Table 1. For demand 1, in step (PL3), the tree, $S_1 = \{V_2, V_3, C_2, A_2, M_{11}, M_{12}, D_4, R_2, W_1, S_1\}$, was chosen randomly, and AC_{S_1} was determined to be 500

Table 2 The results for the solution process of GAMPA and ILOG CPLEX

Method	DC	RC	PC	IC	TC	SC	SP	Total	Solving time (s)	Diff with CPLEX (%)
GAMPA	0	0	27,792	0	26,712	0	0	54,504	15	0.00
CPLEX	0	0	27,792	0	26,712	0	0	54,504	7.88	0.00

The cost component is represented by the following abbreviations: *DC* delay cost, *RC* recycling cost, *PC* processing cost, *IC* inventory holding cost, *TC* transportation cost, *SC* substitution cost, *SP* substitution priority, *Total* total cost

units. The quantity needed to fill demand 1 was 360 units, and thus demand 1 could be filled using S_1 for a total cost of \$26,280. Since $K = 2$, GAMPA tried another tree before proceeding to the next step. In step (PL3), the tree, $S_2 = \{V_1, V_2, V_3, M_{11}, M_{12}, W_1, S_1\}$, was chosen randomly, and AC_{S_2} was determined to be 600 units. Thus, demand 1 could be filled using S_2 for a total cost of \$21,600. GAMPA compared these two results and determined that demand 1 could be completely filled using S_2 for a total cost of \$21,600.

For demand 2, the process was a bit more complicated. In step (PL3), the tree, $S_2 = \{V_1, V_2, V_3, M_{11}, M_{12}, W_1, S_1\}$, was chosen randomly, and AC_{S_2} was determined to be 600 units. The quantity needed to fill demand 2 was 360 units, and thus demand 2 could be filled using S_2 for a total cost of \$21,600 but with a recycling penalty of \$51,600. Again, since $K = 2$, in step (PL3), the tree, $S_3 = \{V_1, V_2, C_1, A_1, M_{11}, M_{12}, D_2, R_1, W_1, S_1\}$, was chosen randomly, and AC_{S_3} was determined to be 500 units. Demand 1 was sold at time bucket 8, so recycled component must be used in order to avoid the recycling penalty. The target recycling ratio was 60 % for P_5, and thus the required recycled component quantity was 216 units, which was smaller than the demand quantity of demand 2 (360 units). Thus, demand 2 could be filled using S_3 for a total cost of \$24,480 but with a recycling penalty of \$0. GAMPA compared these two results and determined that demand 2 could be completely filled using S_3 with a \$0 recycling penalty and a total cost of \$24,480.

Since there were only two demands to plan, GAMPA generated some virtual demands to avoid the recycling penalties for demand 2. So that demand 2 could be sold in time bucket 10, GAMPA generated a virtual demand of $360 \times 60\,\% = 216$ units with a due time bucket of 10. In step (PL3), the tree, $S_3 = \{V_1, V_2, C_1, A_1, M_{11}, M_{12}, D_2, R_1, W_1, S_1\}$, was chosen randomly, and AC_{S_3} was determined to be 500 units. The quantity needed to fill demand 3 (i.e., the virtual demand) was 216 units, and thus demand 3 could be filled using S_3 for a total cost of \$8,424, which is cheaper than the recycling penalty of \$51,600. GAMPA compared the total cost and the recycling penalty and determined that demand 3 (i.e., virtual demand) could be completely filled using S_3 for a total cost of \$8,424, and so this demand was generated in order to avoid the recycling penalty for demand 2.

The first iteration of GAMPA was now finished. To continue the solution process, GAMPA generated another ordered list for these two demands and used the previous calculation methods to obtain another planning result until the stopping condition was met, which in this case was $N = 1$ or $W = 2$. The solution obtained by running GAMPA, shown in Table 2, is the same result as the one obtained with the previous solution process.

This example was used to show that the best result would be a plan that will minimize the delay penalties, minimize substitution, minimize substitution costs, minimize the recycling penalty, and minimize the costs for inventory holding, production, the recycling process and transportation. When planning demand 1, none of the product was sold and thus no recycling penalty was imposed. However, the recycling penalty became an important issue when planning the following demands because the target recycling ratio must be met when products are sold. The recycled component, P_4, was used to fill demand 2—360 units—to avoid the recycling penalty for demand 1. GAMPA generated one virtual demand to avoid the recycling penalty for demand 2 with a quantity of 216 units.

It took GAMPA 15 s to solve this simple problem. The final delay penalty was \$0, both demands met the required target ratio, the number of substitution was 0, the overall recycling penalty was \$0, the substitution cost was \$0, and the costs of production, transportation, the recycling process and inventory holding was \$54,504. For the same problem, ILOG 9.0 constructed an MIP model with 115,142 variables and 118,850 constraints, and produced no optimal solution. CPLEX MIP, using Constraint Programming and a GA heuristic, was able to provide a final solution with the same result as GAMPA. Based on these planning results, GAMPA was able to produce optimal solutions, despite being a heuristic algorithm.

We assumed that there are R demands ($R > 1$ and $DEM_{rp} = D$ for all r and p) with T planning time buckets for P product structures with M items and a supply chain network with V nodes and L links. There are Q substitutions for each item, step (PL3) is repeated K times to obtain a better solution, and the number of generations before stopping is N. For each demand, the complexity of HPFEA when finding a production tree is at most $O(M^2QV + M^2QL)$, which means that each demand needs to go through at most $O((M^2QV + M^2QL) \times T \times (V^{10} \times M^6 + V^{12} \times M^4))$ production trees to determine the final production plan.

Since $T >> 2$, the complexity of planning each demand is $O(M^2 \times Q \times (V+L) \times T \times (V^{10} \times M^6 + V^{12} \times M^4)) = O(V^{11} \times M^8 \times T \times Q + V^{13} \times M^6 \times T \times Q + V^{10} \times M^8 \times T \times Q \times L + V^{12} \times M^6 \times T \times Q \times L)$.

Let Cap_{min} represent Min{Capacity of node i on the minimum cost product tree when the supply chain is empty} and Qty_{total} represent the total quantity of all demands. $T_{Delay} = Qty_{total}/Cap_{min}$ implies that the system needs at most T_{Delay} time buckets to fill all demands if no delay occurs. According to Chern and Huang (2007), in the worst case scenario, each demand must be delayed for at most H periods, where $H = DUE_r + T_{Delay}$. There are R demands, each with a quantity D, which implies that all steps have to be performed at most $O(R \times D)$ times. Thus, the complexity of HPFEA is at most $O(N \times (1/K) \times R \times D \times H \times (V^{11} \times M^8 \times T \times Q + V^{13} \times M^6 \times T \times Q + V^{10} \times M^8 \times T \times Q \times L + V^{12} \times M^6 \times T \times Q \times L))$. Since $P \leq M$ and $Q \leq M$, the complexity of HPFEA is reduced to $O(N \times (1/K) \times R \times D \times H \times (V^{11}M^8T + V^{13}M^6T + V^{10}M^8TL + V^{12}M^6TL))$, $\sim O(R \times D \times H \times V^{13} \times T)$ when $V >> M$, or $\sim O(R \times D \times H \times M^8 \times T)$ when $M >> V$. Compared with the MIP model described in Appendix A, GAMPA is a polynomial heuristic algorithm even in the worst case scenario.

Computational analysis

A prototype based on GAMPA was constructed on a PC server with a Pentium IV 3.5 GHz CPU and 2 GB RAM. This prototype was programmed with Microsoft C#.net on a Microsoft SQL Server 2,000 and run in the Microsoft Windows Server 2,000 environment. CPLEX was used here to provide a lower benchmark of delay penalties, substitutions, recycling penalties and the total costs. Since MIP models with more than 10 demands could not be solved with CPLEX, unlimited capacities were assumed for all nodes to obtain the minimum cost for the lower bounds (Unlimited). The results produced by assuming an unlimited capacity were far better in terms of delay penalties, substitutions, recycling penalties, and the total costs than the results for GAMPA and the MIP models solved by CPLEX, but the solution produced was not feasible.

Section "Problem description and multi-objective functions" presented the simplifications and assumptions needed to generate a production plan that will satisfy all demands while minimizing delay penalties, substitutions, substitution costs, recycling penalties and the costs of production, transportation and inventory holding—all while respecting the capacity limitations and demand deadlines. The accuracy of these simplifications and assumptions were checked through a series of sensitivity analyses. Different parameter conditions were assumed in the various sensitivity analyses. The sensitivities of the objective function and the production plan to the demand, capacity, the required target ratio, substitution, recycling requirements and cost parameters were also examined by these analyses.

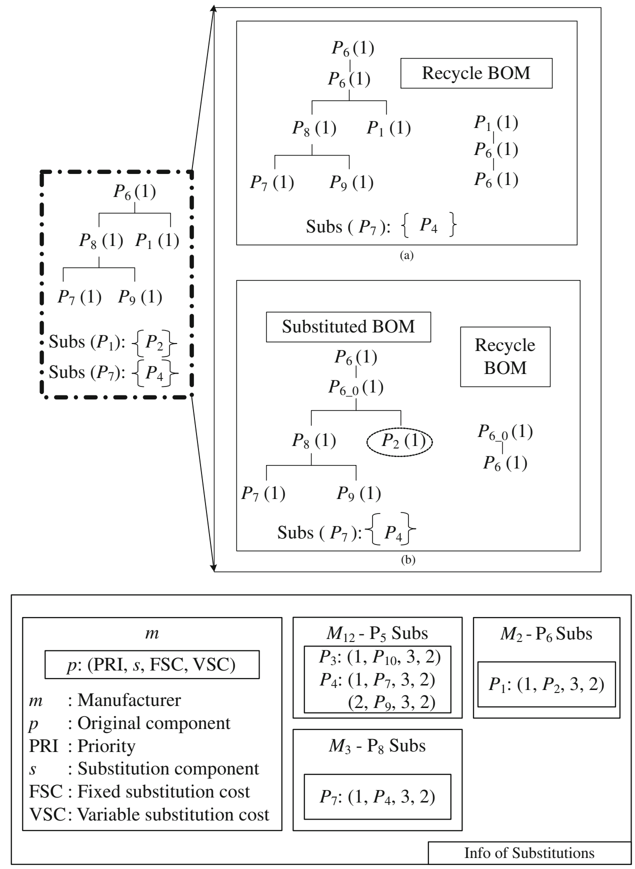

Fig. 3 The BOM used in the computational analysis for scenarios 2, 4, 6 and 8

Like other planning models, the solutions found by CPLEX and GAMPA for the MP problem proposed in this study are fairly robust. Small changes in the parameters do not have much effect on the production plan, let alone the objective functions. To validate the effect of the factors summarized in the production plan and the objective functions, we used a large-scale computational analysis (≥±50 % changes to the parameters) instead of sensitivity analysis. The results obtained from these computational analyses allowed us to make some interesting observations. Since space limitations do not allow all the data obtained from these runs to be presented, a representative subset was chosen to highlight these results and related observations.

We found that three factors affect the solutions of planning problems: capacity, substitution and the target recycling ratio. "Tight capacity" implies that the capacity use rate of each node is greater than 80 %, while "loose capacity" implies that the capacity use rate of each node is less than 70 %. "With substitution" means that the supply chain network carries

Table 3 Scenarios tested

Scenarios	Low target recycling ratio (=0 %)	High target recycling ratio (=60 %)
Without substitute component	Scenario 1 for loose capacity	Scenario 3 for loose capacity
	Scenario 5 for tight capacity	Scenario 7 for tight capacity
With substitute component	Scenario 2 for loose capacity	Scenario 4 for loose capacity
	Scenario 6 for tight capacity	Scenario 8 for tight capacity

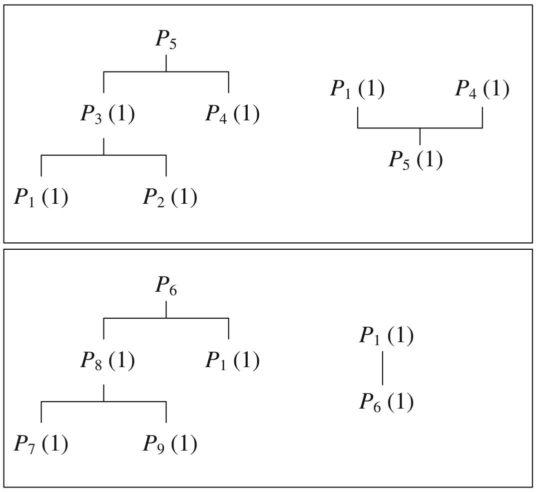

Fig. 4 The BOM used in the computational analysis for scenarios 1, 3, 5 and 7

substitute components, while "without substitution" means that the supply chain network carries no substitute components. A "high target recycling ratio" means the ratio is equal to 60 %, while a "low target recycling ratio" means the ratio is 0 %.

Several scenarios (Figs. 3, 4 and 5) were tested to compare the results produced by GAMPA, CPLEX, and Unlimited (Table 3). The product structure for scenarios 1, 3, 5, and 7 (Fig. 4) indicated no substitute components, while the product structure for scenarios 2, 4, 6, and 8 (Fig. 3) indicated that substitute components exist for the different final products. Figure 5 shows the supply chain network for all scenarios, and Tables 4 and 5 show the node and link information. The major manufacturing nodes are M_{12} and M_2, with a fixed recycling penalty equal to \$30,000 and a variable recycling penalty equal to \$100.

A total of ten demands for each of the eight scenarios (presented in Table 6) were processed. Table 7 shows the system parameters needed for GAMPA in these trials, and Table 8 shows the results of these eight scenarios. Because GAMPA had to preplan demands to avoid recycling penalties and repeated iterations until stopping conditions were met, the GAMPA solution times, ranging from 89 s (1.48 min) to 1,094 s (18.23 min), were longer than most of the CPLEX solution times, which ranged from 100 s (1.67 min) to 237 s (3.95 min), and the Unlimited solution times, which ranged from 86 s (1.43 min) to 90 s (1.5 min). However, CPLEX can only solve MIP models with 10 or less demands and thus was not feasible for solving a real planning problem with 10 or more demands.

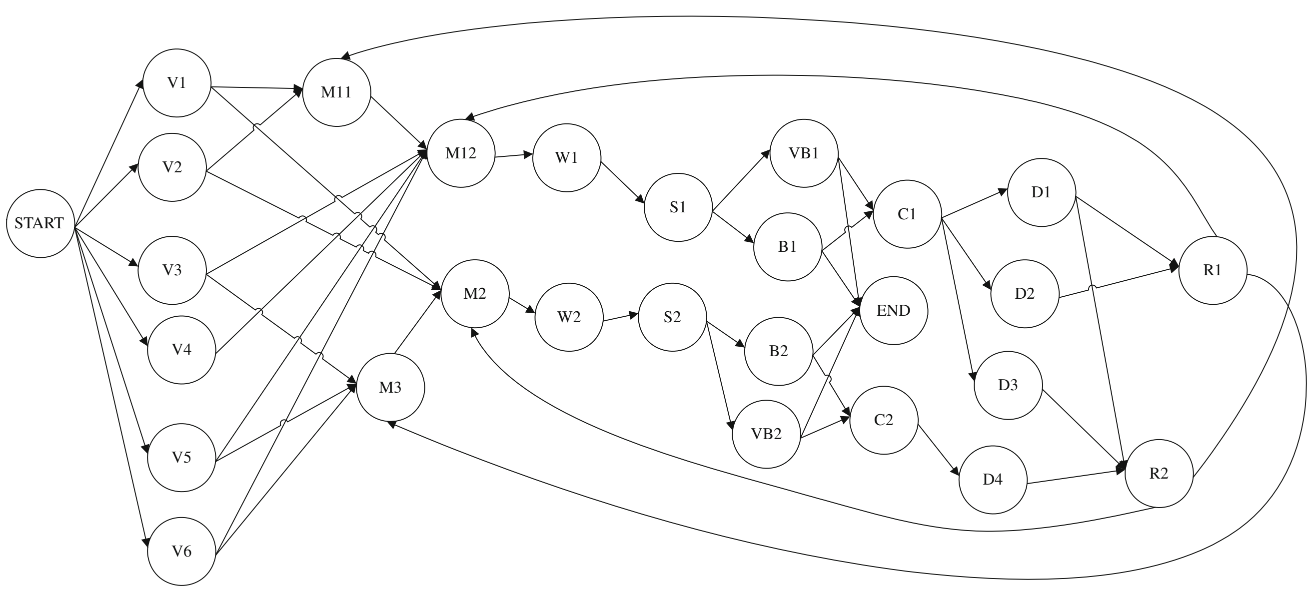

Fig. 5 The supply chain network used in the computational analysis for all scenarios

Table 4 Node information for scenarios tested

Node name	Prod name	P cost	I cost	Capacity (loose)	Capacity (tight)
V_1	P_1	2	3	1,000	300
V_2	P_2	7	3	1,000	300
V_3	P_4	4	3	1,000	250
V_4	P_{10}	4	3	1,000	300
V_5	P_7	4	3	1,000	300
V_6	P_9	4	3	1,000	300
M_{11}	P_3	2	3	1,000	300
M_{12}	P_5	3	3	1,000	300
M_2	P_6	4	3	1,000	300
M_3	P_8	4	3	800	300
W_1	P_5	7	5	600	300
S_1	P_5	5	7	600	300
W_2	P_6	4	6	800	400
S_2	P_6	5	7	800	400
C_1	P_5	0	0	10,000	10,000
C_2	P_6	0	0	10,000	10,000
D_1	P_1,P_4	3	3	1,000	500
D_2	P_4	3	3	600	300
D_3	P_1	3	3	600	300
D_4	P_1	3	3	600	300
R_1	P_4	5	3	500	250
R_2	P_1	5	3	500	250

As expected, GAMPA generated solutions with larger delay penalties, similar substitutions, similar recycling penalties, and similar costs for production, transportation, recycling and inventory holding than CPLEX. The differences between these two algorithms in total costs (delay, substitution, recycling, and production/transportation/inventory holding) ranged from −0.36 to 2.73 %. As the capacity tightened, GAMPA was more thorough in considering the use of substitution or the use of recycled components and thus selected the plans with larger delays, fewer substitutions, or more recycled components that better respected the substitution priorities, recycling penalties, and thus produced smaller overall costs (Scenarios 5 and 6). As the level of substitution became more complicated, GAMPA was more thorough in considering the use of substitution and thus selected the plans with larger delays, fewer substitutions that better respected the substitution priorities and smaller overall costs (Scenario 6). As the target recycling ratio became more complicated, GAMPA was more thorough in considering the use of recycled components and thus selected the plans with larger delays, and fewer recycling penalties that better avoided the fixed recycling penalties (Scenario 7).

The results for the MIP models solved by CPLEX and GAMPA were, on average, equal to 0.69 % different (Table 8). The results for the MIP models solved respectively by CPLEX, Unlimited, and GAMPA were, on average, 15.93 % different between the first two and 16.66 % different between the last two (Table 8). The maximal gap between the lower bound generated by Unlimited and the GAMPA solution was 43.16 % for scenario 8; the maximal gap between the lower bound generated by Unlimited and the

Table 5 Link information for scenarios tested

From node	To node	Product	Lead time	T cost
V_1	M_{11}	P_1	1	3.00
V_2	M_{11}	P_2	1	2.00
V_3	M_{12}	P_4	1	5.00
V_1	M_2	P_1	1	3.00
V_5	M_3	P_7	1	7.00
V_6	M_3	P_9	1	7.00
V_4	M_{12}	P_{10}	1	5.00
V_5	M_{12}	P_7	1	7.00
V_6	M_{12}	P_9	1	7.00
V_2	M_2	P_2	1	2.00
V_3	M_3	P_4	1	5.00
M_{11}	M_{12}	P_3	1	3.00
M_3	M_2	P_8	1	3.00
M_{12}	W_1	P_5	1	9.00
M_2	W_2	P_6	1	3.00
W_2	S_2	P_6	1	3.00
W_1	S_1	P_5	1	8.00
C_1	D_1	P_5	1	4.00
C_1	D_2	P_5	1	4.00
C_1	D_3	P_5	1	4.00
C_2	D_4	P_6	1	5.00
D_1	R_1	P_4	1	3.00
D_1	R_2	P_1	1	3.00
D_2	R_1	P_4	1	3.00
D_3	R_2	P_1	1	3.00
D_4	R_2	P_1	1	3.00
R_2	M_{11}	P_1	1	2.00
R_1	M_{12}	P_4	1	2.00
R_2	M_2	P_1	1	2.00
R_1	M_3	P_4	1	2.00

Table 6 Ten demands for scenarios tested

Demand ID	Demand	Due date	Delay cost	Final product
1	600	8	3	P_5
2	600	8	3	P_6
3	400	9	3	P_5
4	400	9	3	P_6
5	500	10	3	P_5
6	500	10	3	P_6
7	400	10	3	P_5
8	400	11	3	P_6
9	600	11	3	P_5
10	600	12	3	P_6

Table 7 GAMPA system parameters for scenarios tested

Term	Default value with substitution	Default value without substitution
# of generations	10	10
# of successive identical sequences	4	4
Mutation rate	0.2	0.2
# of tries before delay	14	8
# of runs	14	8
Initial recycling capacity	5,000	5,000
Goal setting	(1) Min. total delay cost; (2) min. recycling cost; (3) min. substitution cost; (4) min. substitution priority; (5) min. total processing cost	

CPLEX solution was 42.72 % for scenario 8 too. The results assuming the unlimited capacity solved by Unlimited were far less than the results of the MIP models solved by CPLEX and GAMPA. Therefore, we conclude that the GAMPA produces a feasible, good quality solution because it is generally relatively closer to the optimal solution obtained by CPLEX than the infeasible solution obtained by Unlimited, assuming unlimited capacity.

As mentioned before, the MIP model presented in Appendix A is not solvable for master planning problems with more than 10 demands. Even if the MIP model is solvable, the optimal solution may not lead to a tighter lower bound because of the multi-stage optimization process for the five subsequent objective functions. Using the multi-stage optimization process, the first objective function is minimized, then the second, the third, and so on. Sometimes, a GAMPA solution performs better than the MIP solution for the second or third objective function because the minimum is not achieved for the first objective function. For example, the MIP solution achieves a smaller delay cost but a higher substitution cost (Scenario 6) or a smaller delay cost but a higher recycling penalty (Scenario 7) than the solutions from GAMPA.

In reality, company management not only wants to avoid delays by using a prioritized substitute but also wants to use a substitute whenever the total cost can be minimized or a recycled component whenever the recycling penalty can be avoided. Therefore, the five objective functions are considered simultaneously with different accounts. However, the MIP model does not have flexibility to switch the order of these objective functions in the solution process for the same problem. For each problem, the order of these objective functions is set before beginning to solve the MIP model. Therefore, it may sometimes obtain an optimal solution that is not exactly what the management had in mind.

GAMPA, on the other hand, has the flexibility to take these objective functions into account by evaluating the substitution cost, the recycling penalty and the inventory holding cost to determine if switching to the substitute items is necessary, if replacing items based on their priorities is desirable or if using recycled components is cost effective to avoid the recycling penalty. Therefore, GAMPA may produce a solution that is not optimal (compared to the MIP solution) but may be more acceptable to the management.

To show the time efficiency and effectiveness of GAMPA, the 10 demands in scenario 8 were repeated n times ($n = 10$, 20, 30, 40, and 50), with 11 time buckets added to the due dates of all the demands at each iteration. Since MIP models with more than 10 demands could not be solved with CPLEX, unlimited capacities were assumed for all nodes and solved by GAMPA to obtain the minimum cost for the lower bounds. The gap between the Unlimited results and the results produced by GAMPA reflects the maximal difference between the optimal solution and the GAMPA solution. (These results are presented in Fig. 6.) The Unlimited solution times for planning 100–500 demands ranged from 4,824 s (1 h 20 min 24 s) to 22,248 s (6 h 10 min 48 s). GAMPA took from 7,248 s (2 h 3 min 48 s) to 43,371 s (12 h 3 min 51 s) to complete the same task because the algorithm pre-planned demands or created virtual demands to avoid the recycling penalties, which made the planning process much more complicated. The solution times of GAMPA increased linearly as the number of demands incremented. The gaps between lower bounds of the total costs and the results with GAMPA increased linearly as the number of demands incremented.

A large example with 2,000 demands (with 200, 300, 400, and 600 units distributed uniformly, in which 50 % of the demands are for product X, 30 % for product T, and 20 % for product A), a seven-level supply chain network and product structures for three final products with a recycling process (Figs. 7, 8) was also tested. The data came from a large PC and NB assembly plant based in Taiwan and is listed in Table 9. The target recycling ratio was assumed to be 50 % for all final products; the fixed recycling penalty was \$1 million, and the variable recycling penalty was \$1,000. The definitions of the required system parameters for GAMPA is listed in Table 10. GAMPA took 42,540 s (11 h 49 min) to plan all

Table 8 Planning results for scenarios tested

Scenario	Method	DC	FRC	VRC	FSC	VSC	SP	PC	IHC	TC	Total	Time	Diff (%)	Diff with unlimited (%)
1 (LLN)	GAMPA	0	0	0	0	0	0	145,900	4,400	143,500	293,800	1 min 29 s	0.14	1.88
	CPLEX	0	0	0	0	0	0	148,380	5,000	140,000	293,380	1 min 40 s		1.73
	Unlimited	0	0	0	0	0	0	148,380	0	140,000	288,380	1 min 26 s		
2 (LLY)	GAMPA	0	0	0	0	0	0	145,700	4,400	143,400	293,500	3 min 13 s	0.04	1.78
	CPLEX	0	0	0	0	0	0	148,380	5,000	140,000	293,380	1 min 40 s		1.73
	Unlimited	0	0	0	0	0	0	148,380	0	140,000	288,380	1 min 26 s		
3 (LHN)	GAMPA	0	0	0	0	0	0	188,700	4,340	183,640	376,680	9 min 18 s	2.68	4.87
	CPLEX	0	0	0	0	0	0	187,840	6,300	172,700	366,840	1 min 41 s		2.13
	Unlimited	0	0	0	0	0	0	185,880	600	172,700	359,180	1 min 30 s		
4 (LHY)	GAMPA	0	0	0	0	0	0	189,100	4,460	183,300	376,860	8 min 29 s	2.73	4.92
	CPLEX	0	0	0	0	0	0	187,840	6,300	172,700	366,840	1 min 41 s		2.13
	Unlimited	0	0	0	0	0	0	185,880	600	172,700	359,180	1 min 30 s		
5 (TLN)	GAMPA	6,750	0	0	0	0	0	157,000	20,500	155,100	339,350	12 min 53 s	−0.76	17.67
	CPLEX	6,300	0	0	0	0	0	161,800	22,750	151,100	341,950	1 min 44 s		18.58
	Unlimited	0	0	0	0	0	0	148,380	0	140,000	288,380	1 min 26 s		
6 (TLY)	GAMPA	3,450	0	0	9	800	3	152,500	36,950	152,300	346,009	9 min 18 s	−0.36	19.98
	CPLEX	2,100	0	0	6	1,200	2	157,170	39,000	147,800	347,276	2 min 38 s		20.42
	Unlimited	0	0	0	0	0	0	148,380	0	140,000	288,380	1 min 26 s		
7 (THN)	GAMPA	7,200	90,000	28,000	0	0	0	177,044	23,390	173,562	499,196	18 min 14 s	0.74	38.98
	CPLEX	6,300	90,000	33,000	0	0	0	176,386	26,096	163,754	495,536	1 min 46 s		37.96
	Unlimited	0	0	0	0	0	0	185,880	600	172,700	359,180	1 min 30 s		
8 (THY)	GAMPA	2,400	90,000	36,000	18	1,660	6	171,050	42,350	170,740	514,218	16 min 8 s	0.31	43.16
	CPLEX	2,100	90,000	30,000	9	1,380	3	176,670	46,500	165,950	512,609	3 min 57 s		42.72
	Unlimited	0	0	0	0	0	0	185,880	600	172,700	359,180	1 min 30 s		

The cost component is represented by the following abbreviations: *DC* delay cost, *FRC* fixed recycling cost, *VRC* variable recycling cost, *FSC* fixed substitution cost, *VSC* variable substitution cost, *SP* substitution priority, *PC* processing cost, *IHC* inventory holding cost, *TC* transportation cost, *Total* total cost

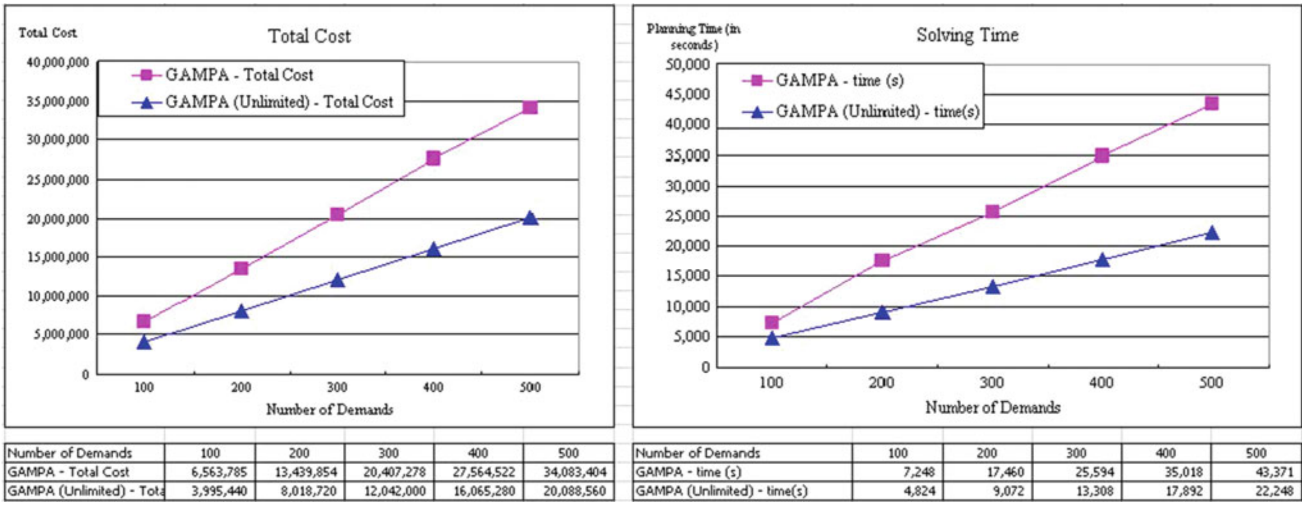

Fig. 6 Results for GAMPA and unlimited capacity

Table 9 Cost information for a real-world case

Price or cost per unit/NB type	X	T	A
Retail price	40,000	50,000	60,000
Retail profit (18 %)	7,200	9,000	10,800
Materials cost (54 %)	21,600	27,000	32,400
Operational cost (10 %)	4,000	5,000	6,000
Labor cost (18 %)	7,200	9,000	10,800
Inventory holding cost (% of material cost)	1	1	1
Fixed substitution cost (multiple of operation cost)	100	100	100
Variable substitution cost (% difference of material cost)	100	100	100

Table 10 GAMPA system parameters for a real-world case

Term	Default value
# of generations	10
# of successive identical sequence	4
# of tries before delay	5
# of runs	4
Initial recycling capacity	5,000
Goal setting	(1) Min. total delay cost; (2) min. recycling cost; (3) min. substitution cost; (4) min. substitution priority; (5) min. total processing cost

Table 11 Results for the real-world case

Method	DC	FRC	VRC	FSC	VSC	SP	PC	IHC	TC	Total	Time (s)
GAMPA	0	0	0	2,600,000	864,000	6	19,317,755,000	56,277,200	9,752,541,000	29,130,037,200	42,540
Unlimited	0	0	0	0	0	0	19,318,079,000	56,277,200	9,752,541,000	29,126,897,200	42,050

The cost component is represented by the following abbreviations: *DC* delay cost, *FRC* fixed recycling cost, *VRC* variable recycling cost, *FSC* fixed substitution cost, *VSC* variable substitution cost, *SP* substitution priority, *PC* processing cost, *IHC* inventory holding cost, *TC* transportation cost, *Total* total cost

Table 12 GAMPA solutions for scenarios tested

Scenario	State	Order sequence										Diff (%)	Generation stop
1	Final	1	2	3	4	6	5	7	9	8	10	0	1
	Initial_rule	1	2	3	4	5	6	7	9	8	10		
2	Final	2	1	6	3	5	4	7	9	10	8	**20**	6
	Initial_rule	1	2	3	4	5	6	7	9	8	10		
3	Final	1	2	4	3	6	7	5	9	8	10	**20**	9
	Initial_rule	1	2	3	4	5	6	7	9	8	10		
4	Final	1	2	4	3	6	8	5	7	9	10	**30**	4
	Initial_rule	1	2	3	4	5	6	7	9	8	10		
5	Final	2	8	1	3	4	5	6	7	9	10	**60**	4
	Initial_rule	1	2	3	4	5	6	7	9	8	10		
6	Final	2	1	3	4	6	5	7	9	8	10	0	3
	Initial_rule	1	2	3	4	5	6	7	9	8	10		
7	Final	2	1	4	3	5	6	7	9	8	10	0	3
	Initial_rule	1	2	3	4	5	6	7	9	8	10		
8	Final	1	2	4	3	6	5	7	8	9	10	0	1
	Initial_rule	1	2	3	4	5	6	7	9	8	10		

The numbers are in bold face because they are larger than 0

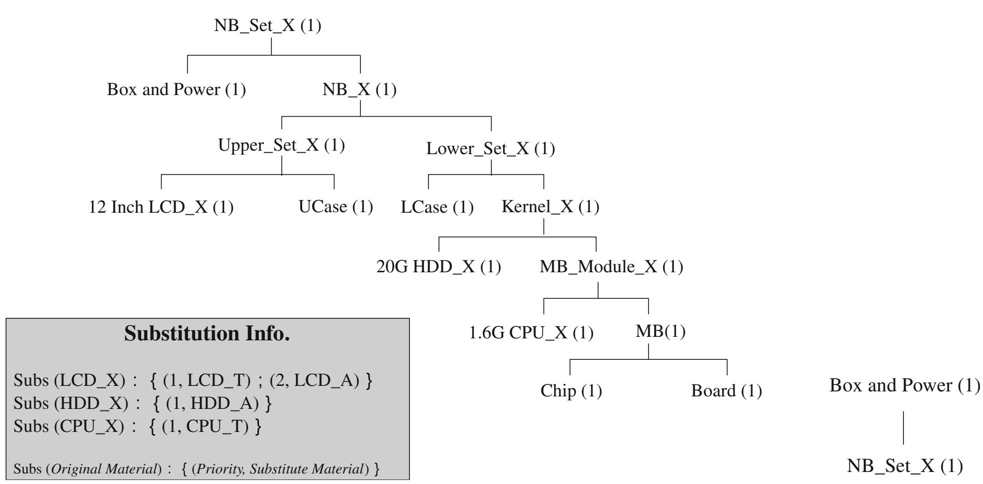

Fig. 7 A product structure for a real-world case

2,000 demands. Table 11 provides the results, and they are fairly good. In a real-world business situation, the number of demands is often more than 1,000 per month, and the planning time horizon is more than 6 months, with lead times ranging from 1 day to several months. The scale of an MP problem for supply chains with substitution and a recycling process is often gigantic and impossible to solve using an MIP model, but GAMPA can solve this kind of problem in reasonable time as demonstrated above.

Though GAMPA appears to have many advantages for solving MP problems with substitutions and a recycling process, notably in terms of problem scales and planning results, GAMPA may miss the optimal solutions for the following reasons:

1. Repeated fitness function evaluation for complex problems is the most prohibitive and limiting segment of GAMPA. Even with approximation model in GAMPA, a real-world problem still required several hours to complete the search and evaluation algorithm.
2. The "better" solution generated by GAMPA is only in comparison to others. As a result, the stopping criterion in GAMPA is not clearly defined. In some problems, GAMPA may have a tendency to converge towards local optima, or even arbitrary points, rather than the global optimum of the problem. This difficulty may be alleviated by increasing the rate of mutation, although, if this is done, it will take much longer for GAMPA to find solutions.

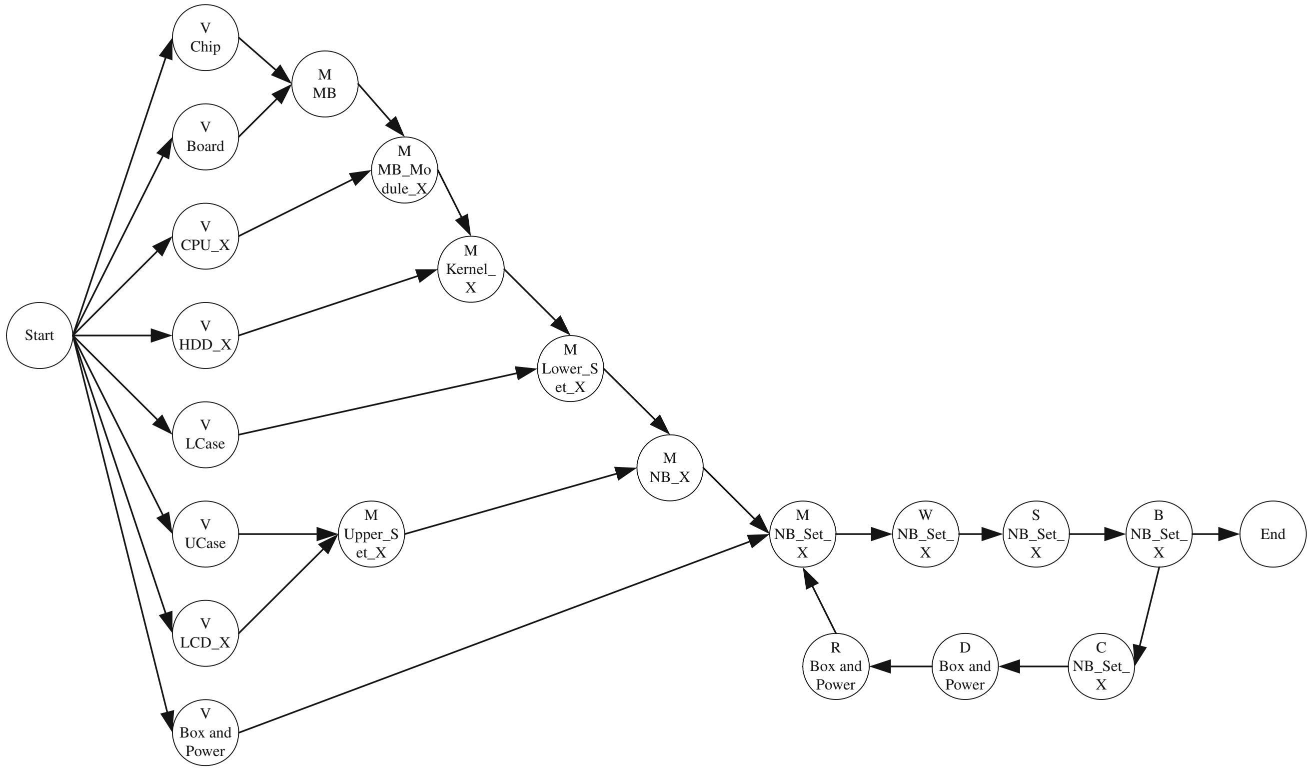

Fig. 8 A supply chain network for a real-world case

3. For the master planning problems considered in this study, CPLEX optimization algorithm may find better solutions than genetic algorithms, given the same amount of computation time. However, MIP models with more than 10 demands cannot be solved with CPLEX. GAMPA is better suited to solving master planning problems with more than 10 demands, and this means GAMPA is better for real-world problems because, as mentioned above, the number of demands is often more than 1,000 per month.
4. A very small mutation rate may lead to genetic drift, while a very high mutation rate may lead to a loss of good solutions. It is worth fine-tuning the system parameters of GAMPA (e.g., mutation probability, population size, and/or number of generations before stopping) to find reasonable settings for the problem class being worked on.

Often, GAMPA can rapidly locate good solutions, thanks to the initial rule-based solution generation method, even for large search spaces. As shown in Table 12, four of the eight scenarios had the same final solution as the initial rule-based solution and the remaining four are similar to the initial solution, except only one with 60 % difference. These results prove that GAMPA's rule-based initial solution generator can improve GAMPA's efficiency in terms of solution time and the objectives.

It is true that sometimes GAMPA is not able to produce optimal solutions, compared to the results from the MIP model. Nonetheless, GAMPA is able to solve large problems with tolerable efficiency because, when the number of demands are too large (i.e., more than 10), the master planning problem with substitution and a recycling process becomes insolvable for CPLEX but is still manageable for GAMPA.

Conclusion

This study proposes a heuristic algorithm called the GA-based Master Planning Algorithm (GAMPA) for solving a master planning problem for a supply chain network with multiple final products, substitutions, and a recycling process. The recycling process creates a loop in the supply chain network, as well as in the product structure trees. The objectives of GAMPA are (1) minimizing delay costs, (2) minimizing recycling penalties, (3) minimizing substitutions while respecting substitution priorities, (4) minimizing substitution costs, and (5) minimizing the costs of production, transportation, and inventory holding—while respecting capacity limitations and due date constraints. Mixed Integer Programming (MIP) is usually used to solve these types of problems, but the MIP model is unsolvable due to the complexity of the variables and constraints. In addition, solving MIP models always takes a long time and requires computers with enormous computing capacities. To better improve the efficiency and effectiveness of the planning procedure, this study proposes GAMPA for solving such multi-objective planning problems for supply chains with substitution and a recycling process. Despite its advantages, GAMPA currently assumes the recycling pattern is a stochastic process with a multinomial distribution. Therefore, the solutions from the MIP models or GAMPA are optimal or near optimal planning results based on the average recycling returns. More studies are needed to investigate the impact of recycling pattern variations on the planning processes, and contingence policies are needed to deal with these situations.

Acknowledgments This research was sponsored by the National Science Committee of Taiwan, under the project number NSC 98-2416-H-002-012-MY2.

Appendix A

MIP model formulation

The following indices and parameters are defined prior to the construction of a MIP model.

Indices

i, j, and g are nodes; r represents demands, with R being equal to the total number of demands; t is the time bucket, with T being equal to the total number of planning time buckets; m, n, w, and s are respectively parent items and components; and p stands for final products, with P being equal to the total number of final products.

Parameters

V: The set of nodes in a supply chain network. Specifically, the *Start* node and the *End* node are the source (origin) and sink (destination), respectively.

V': The set of nodes in a supply chain network, except the *Start* node and the *End* node.

V^V: The set of raw materials supplying the nodes in a supply chain network.

V^M: The set of manufacturing nodes in a supply chain network.

V^P: The set of distribution nodes in a supply chain network.

V^B: The set of customer nodes in a supply chain network.

V^{VB}: The set of virtual customer nodes in a supply chain network.

V^C: The set of recycled component collection nodes in a supply chain network.

V^D: The set of recycled component disassembly nodes in a supply chain network.

V^R: The set of recycled component reprocessing nodes in a supply chain network.

L: The set of links in a supply chain network.

M: The set of all raw materials, components, semi-finished products, and final products.

Q: The set of all virtual final products.

Q^p: The set of all virtual final products for final product p.

SBS_m: The set of substitutions for item m.

DUE_{pr}: Due date (in time buckets) for final product p of demand r.

DEM_{pr}: Demand quantity for final product p of demand r.

CX_{it}: The capacity limit of node i at time bucket t, where $i \in V^V \cup V^M \cup V^D$.

CS_{it}: The capacity limit of node i at time bucket t, where $i \in V^P \cup V^R$.

Cx_{in}: Capacity needed at node i when producing one unit of n, where $i \in V^V \cup V^M \cup V^D$.

Cs_{in}: Capacity needed at node i when distributing one unit of n, where $i \in V^P \cup V^R$.

LT_{ijn}: Lead Time (in time buckets) needed to transfer item n on link (i, j), where $(i, j) \in L$.

TC_{ijn}: The incremental unit cost for shipping item n from node i to node j, where $(i, j) \in L$.

PC_{in}: The incremental unit processing cost for item n at node i, where $i \in V^V \cup V^M \cup V^D$.

QC_{in}: The incremental unit processing cost for item n at node i, where $i \in V^P \cup V^R$.

HC_{in}: The unit holding cost for item n at node i and $i \in V$.

DC_{pr}: The unit delay penalty per unit time bucket for item p of demand r, where $p \in$\{Final Items\}.

PAR_m: The set of parent items for item m.

RCH_m: The set of recycled items for recycled item m.

PRD_i: The set of items produced on node i, where $i \in V$.

COM_i: The set of components shipped to node i, where $i \in V^M \cup V^D$.

USE_{mn}: Quantity of item m needed per unit of item n, where $n \in Par_m$.

$RUSE_{mn}$: Quantity of recycled item m reprocessed from item n per unit, where $n \in RCH_m$.

RF_i: The fixed recycling penalty tag for manufacturer i, where $i \in V^M$. If $RF_i = 1$, the fixed recycling penalty is applied to manufacturer i. Otherwise, no fixed recycling penalty is applied to manufacturer i.

RV_i: The variable recycling penalty tag for manufacturer i, where $i \in V^M$. If $RV_i = 1$, the variable recycling penalty is applied to manufacturer i. Otherwise, no variable recycling penalty is applied to manufacturer i.

$RATE_p$: The target recycling ratio for final product p where $p \in$\{Final Items\}.

PF_{ip}: The fixed recycling penalty to charge to manufacturer i (where $i \in V^M$) for final product p when $RF_i = 1$ and the $RATE_p$ is not met.

PV_{ip}: The variable recycling penalty to charge to manufacturer i (where $i \in V^M$) for final product p when RV_i =1 and the $RATE_p$ is not met.

$PROB_{pmt}$: The multinomial probability distribution that final product p is recycled and produced component m at period t, where $\sum_{t=1}^{T} PROB_{pmt} = 1$.

PRI_{mns}: The priority of substitute s when used to replace item m in the production of item n, where $s \in SBS_m$.

SFC_{inms}: The fixed cost for replacing item m with substitute s in the production of item n at node i, where $s \in SBS_m$.

SVC_{inms}: The unit cost for replacing item m with substitute s in the production of item n at node i, where $s \in Sub_m$.

I_{inp0}: The beginning inventory of item n for final product p of demand r at node i in time bucket 0, where $i \in V - V^C - V^D$.

IU_{n0}: The beginning inventory of item n at node i in time bucket 0, where $i \in V^C \cup V^D$.

Small_m: A very small positive number.

Big_M: A very large positive number.

Decision variables

X_{inprt}: The production quantity of item n for final product p of demand r at node i in time bucket t.

Z_{int}: The quantity of recycled item m reprocessed at node i in time bucket t.

S_{ijnprt}: The quantity of item n needed for final product p of demand r to be shipped from node i to node j in time bucket t, where $j \in V - V^C - V^D$.

U_{ijnt}: The quantity of item n to be shipped from node i to node j in time bucket t, where $j \in V^C \cup V^D$.

I_{inprt}: The ending inventory quantity of item n for final product p of demand r at node i in time bucket t, where $j \in V - V^C - V^D$.

IU_{int}: The ending inventory quantity of item n at node i in time bucket t, where $j \in V^C \cup V^D$.

WL_{ipt}: The deficit quantity of final product p at node i in time bucket t to meet the target recycling ratio, $RATE_p$.

WM_{ipt}: The surplus quantity of final product p at node i in time bucket t to meet the target recycling ratio, $RATE_p$.

YR_{ipt}: A binary variable that indicates if the target recycling ratio for final product p at node i in time bucket t is satisfied or not. If the $Recyle\_Ratio_{ipt} > RATE_p$, $YR_{ipt} = 1$. Otherwise, $YR_{ipt} = 0$.

$SUB_{inmsprt}$: The quantity of a substitute s needed to replace item m in the production of item n to fill demand r for final product p at node i in time bucket t.

$YS_{inmsprt}$: A binary variable that indicates the situation of the substitute s, which replaces item m in the production of item n to fill demand r of final product p at node i in time bucket t. If $SUB_{inmsprt} > 0, YS_{inmsprt} = 1$. Otherwise, $YS_{inmsprt} = 0$.

The constraints

(a) $\sum_{p=1}^{P} \sum_{r=1}^{R} \sum_{n \in PRD_i} Cx_{in} X_{inprt} \le CX_{it} \forall i \in V^V \cup V^M, 1 \le t \le T$

(b) $\sum_{n \in COM_i} Cx_{in} Z_{int} \le CX_{it} \forall i \in V^D, 1 \le t \le T$

(c) $\sum_{p=1}^{P} \sum_{r=1}^{R} \sum_{i \in V} \sum_{n \in PRD_i} Cs_{in} S_{ijnpr(t-LT_{gim})} \le CS_{jt} \forall j \in V^P \cup V^R, LT_{gim} \le t \le T$

(d) $\sum_{i \in V} \sum_{t=LT_{iEndp}}^{T} S_{ijnpr(t-LT_{iEndp})} = DEM_{pr} \forall p, r$

(e) $IU_{jn(t-1)} + \sum_{m \in RCH_n \cap COM_j} Z_{jmt} \times RUSE_{mn} - \sum_{p=1}^{P} \sum_{r=1}^{R} \sum_{(j,k) \in L} S_{jknprt} = IU_{jnt} \forall j \in V^D, n = PRD_j, 1 \le t \le T$

(f) $IU_{im(t-1)} + \sum_{(h,i) \in L} U_{him(t-LT_{him})} - Z_{imt} = IU_{imt} \forall i \in V^D, m \in COM_i, LT_{him} \le t \le T$

(g) $IU_{in(t-1)} + \sum_{(h,i) \in L} U_{hin(t-LT_{hin})} - \sum_{(i,j) \in L} U_{ijnt} - \sum_{p=1}^{P} \sum_{r=1}^{R} \sum_{(i,k) \in L} S_{iknprt} = IU_{int}\; i \in V^C, n \in PRD_i, LT_{hin} \le t \le T$

(h) $\sum_{p=1}^{P} \Big(I_{inpr(t-1)} + X_{inprt} - \sum_{(j,k) \in L} S_{jknprt} + \sum_{(i,j) \in L} S_{ijnpr(t-LT_{ijn})} \Big) = \sum_{p=1}^{P} I_{jnprt} \forall i \in V^V \cup V^M, m = PRD_i, r, LT_{ijn} \le t \le T$

(i) $\sum_{p=1}^{P} \Big(I_{impr(t-1)} + \sum_{(h,i) \in L} S_{himpr(t-LT_{him})} - \sum_{n \in PAR_m} X_{inprt} \times USE_{mn} + \sum_{n \in PRD_i} \sum_{q \in SBS_m} SUB_{inmqprt} - \sum_{n \in PRD_i} \sum_{m \in SBS_s} SUB_{inqmprt} \Big) = \sum_{p=1}^{P} I_{imprt} \forall i \in V^M, m \in COM_i, r, LT_{him} \le t \le T$

(j) $\sum_{p=1}^{P} \Big(I_{jnpr(t-1)} + \sum_{(i,j) \in L} S_{ijnpr(t-LT_{ijn})} - \sum_{(i,j) \in L} S_{jknprt} \Big) = \sum_{p=1}^{P} I_{jnprt} \forall j \in V^P \cup V^R, n \in PRD_i, r, LT_{ijn} \le t \le T$

(k) $\sum_{(j,k) \in L} U_{jknt} - \sum_{p=1}^{P} \sum_{r=1}^{R} \sum_{(i,j) \in L} S_{ijnpr(t-LT_{ijn})} = 0 \forall j \in V^{VB}, n \in PRD_j, LT_{ijn} \le t \le T$

(l) $\sum_{(j,k) \in L} U_{jknt} - \sum_{p=1}^{P} \sum_{r=1}^{R} \sum_{(i,j) \in L} \sum_{d=1}^{t} S_{ijppr(t-LT_{ijp}-d)} \times PROB_{npd} \times USE_{np} = 0 \; \forall j \in V^B - V^{VB}, n \in PRD_j, LT_{ijn} + d \le t \le T$

(m) $\sum_{r=1}^{R} \Big(\sum_{(j,k) \in L} S_{jkpprt} \times RATE_p - \sum_{(h,i) \in L} S_{hinpr(t-LT_{hin})} \times \frac{1}{USE_{np}} \Big) - Big\_M \times YR_{ipt} \le 0 \; \forall i \in V^M, k \in V^B, h \in V^R, j, n, p \in Q_p, LT_{hin} \le t \le T$

(n) $\sum_{r=1}^{R} \Big(\sum_{(j,k) \in L} S_{jkpprt} \times RATE_p - \sum_{(h,i) \in L} S_{hinpr(t-LT_{hin})} \times \frac{1}{USE_{np}} \Big) - Small\_M \times YR_{ipt} \ge 0 \; \forall i \in V^M, k \in V^B, h \in V^R, j, n, p \in Q_p, LT_{hin} \le t \le T$

(o) $WM_{ipt} - WL_{ipt} = \sum_{r=1}^{R} \left(\sum_{(j,k) \in L} S_{jkpprt} \times RATE_p - \sum_{(h,i) \in L} S_{hinpr(t-LT_{hin})} \times \frac{1}{USE_{np}} \right)$ $\forall i \in V^M, k \in V^B, h \in V^R, j, n, p \in Q_p, LT_{hin} \leq t \leq \mathrm{T}$

(p) $\sum_{r=1}^{R} \left(\sum_{(h,i) \in L} S_{hinpr(t-LT_{hin})} \times \frac{1}{USE_{np}} - X_{inprt} \right) \leq 0 \, \forall i \in V^M, h \in V^R, n, p \in Q_p, LT_{hin} \leq t \leq T$

(q) $Small\_m * YS_{inmsprt} \leq SUB_{inmsprt} \, \forall i \in V^M, m, r, t, p, n \in PRD_i \cap PAR_m, s \in SBS_m$

(r) $Big\_M * YS_{inmsprt} \geq SUB_{inmsprt} \, \forall i \in V^M, m, r, t, p, n \in PRD_i \cap PAR_m, s \in SBS_m$

(s) $I_{inprt} = 0 \quad \forall i \in V^B, p, r, t < DUE_{pr}$

(t) $S_{ijnpr0} = 0 \quad \forall j \in V^V \cup V^M \cup V^P \cup V^B \cup V^R, (i, j) \in L, p, n, r$

(u) $U_{ijn0} = 0 \quad \forall j \in V^C \cup V^D, \ (i, j) \in L, \ n$

(v) $I_{inprt} \geq 0 \quad \forall i \in V^V \cup V^M \cup V^P \cup V^B \cup V^R, p, n, r, t$

(w) $X_{inprt} \geq 0 \quad \forall i \in V^V \cup V^M, p, n \in PRD_i, r, t$

(x) $U_{ijnt} \geq 0 \quad \forall j \in V^C \cup V^D, \ (i, j) \in L, \ n, t$

(y) $IU_{imt} \geq 0 \quad \forall i \in V^C \cup V^D, m, t$

(z) $Z_{imt} \geq 0 \quad \forall i \in V^D, m \in PRD_i, t$

(aa) $S_{ijnprt} \geq 0 \quad \forall j \in V^V \cup V^M \cup V^P \cup V^B \cup V^R, (i, j) \in L, p, n, r, t$

(bb) $WM_{ipt} \geq 0 \quad \forall i \in V^M, p, t$

(cc) $WL_{ipt} \geq 0 \quad \forall i \in V^M, p, t$

(dd) $SUB_{inmsprt} \geq 0 \forall i \in V^M, m, r, t, p, n \in PRD_i \cap PAR_m, s \in SBS_m$

(ee) $YR_{ipt} \in \{0, 1\} \forall i \in V^M, t, p$

(ff) $YS_{inmsprt} \in \{0,1\} \, \forall i \in V^M, m, r, t, p, n \in PRD_i \cap PAR_m, s \in SBS_m$

Constraints (a) and (b) represent the production capacity limitations for all components at raw material supply nodes, manufacturing nodes, disassembly nodes or reprocessing nodes in each time bucket. Constraint (c) shows the distribution capacity limitations for all components at the distribution nodes in each time bucket. Constraint (d) requires the total quantity of transported final products at the customer node after the due date to be equal to the total demand quantity for each demand. Constraints (e)–(j) are the inventory balancing equations for each semi-finished, final product or recycled item at each raw material supply node or manufacturing node in each time bucket. Constraint (k) and (l) are the inventory balancing equations for each recycled item at each collection, disassembly or reprocessing node in each time bucket. Constraints (m) and (n) set the values of binary variables for the fixed and variable recycling penalties at the manufacturing nodes in each time bucket. Constraint (o) sets the inventory at the customer node for demand r equal to 0 after due date time bucket. Constraint (p) limits the recycled material to be used in production at the manufacturing node for each recycled component in the same time bucket in which the component is reprocessed. Constraints (q) and (r) fix the relationship between the substitution quantity and the binary variable. Constraint (s) sets the initial inventory at each node in time bucket 1 equal to 0. Constraint (t) sets the initial shipping quantity at each node in time bucket 1 equal to 0. Constraint (u) sets the initial recycled component quantity at each node in time bucket 1 equal to 0. Constraints (v)–(dd) show the non-negative requirement for each decision variable. Constraints (ee)–(ff) show the binary requirement for each binary decision variable.

Multiple objective functions

Five objective functions must be accomplished: (1) Minimize the total delay cost; (2) Minimizing substitutions while respecting substitution priorities; (3) Minimize the substitution cost; (4) Minimize the recycling penalty; and (5) Minimize the total cost of production, transportation and inventory holding.

Objective 1: Minimize the total delay cost, *TDC*.

$$\mathrm{Min}\, TDC = \sum_{r=1}^{R} \sum_{p=1}^{P} \sum_{r=1} DC_{pr} \times \sum_{t=DUE_{pr}}^{T} \left[DEM_{pr} - I_{ENDpprt} \right]$$

Objective 2: Minimizing substitutions while respecting substitution priorities, *SBP*.

$$\mathrm{Min}\, SBP = \sum_{r=1}^{R} \sum_{p=1}^{P} \sum_{m=1}^{M} \sum_{t=1}^{T} \sum_{n \in PRD_i \cap PAR_m} \sum_{s \in SBS_m} \sum_{i \in V^M} PRI_{nms} YS_{inmsprt}$$

Objective 3: Minimize the substitution cost, *SBC*.

$$\mathrm{Min}\, SBC = \sum_{r=1}^{R} \sum_{p=1}^{P} \sum_{m=1}^{M} \sum_{i \in V^M} \sum_{t=1}^{T} \sum_{n \in PRD_i \cap PAR_m} \sum_{s \in SBS_m} SFC_{inmq} YS_{inmsprt} + \sum_{r=1}^{R} \sum_{p=1}^{P} \sum_{m=1}^{M} \sum_{i \in V^M} \sum_{t=1}^{T} \sum_{n \in PRD_i \cap PAR_m} \sum_{s \in SBS_m} SVC_{inmq} SUB_{inmsprt}$$

Objective 4: Minimize the recycling penalty, *RCP*.

$$\text{Min } RCP = \sum_{p=1}^{P} \sum_{i \in V^M} \sum_{t=1}^{T} RF_i \times PF_{ip} \times YR_{ipt} + \sum_{p=1}^{P} \sum_{i \in V^M} \sum_{t=1}^{T} RV_i \times PV_{ip} \times WL_{ipt}$$

Objective 5: Minimize the total cost of production, transportation and inventory holding, *PTI*.

$$\begin{aligned}\text{Min } PTI = & \sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{t=1}^{T}\sum_{i \in V^V \cup V^M} \sum_{n \in PRD_i} PC_{in} \times X_{inprt} \\ & + \sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{t=LT_{ijn}}^{T}\sum_{i \in V^P} \sum_{n \in PRD_i}\sum_{(i,j)\in L} QC_{in} \times S_{ijnpr(t-LT_{ijn})} \\ & + \sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{t=1}^{T}\sum_{i \in V^D} \sum_{m \in PRD_i} PC_{im} \times Z_{imt} \\ & + \sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{t=LT_{ijn}}^{T}\sum_{i \in V^R} \sum_{n \in PRD_i} QC_{in} \times S_{ijnpr(t-LT_{ijn})} \\ & + \sum_{(i,j)\in L}\sum_{n \in PRD_i}\sum_{t=LT_{ijn}}^{T} QC_{in} U_{ijn(t-LT_{ijn})} \\ & + \sum_{(i,j)\in L}\sum_{n \in PRD_i} TC_{ijn} \left(\sum_{t=1}^{T}\left(\sum_{p=1}^{P}\sum_{r=1}^{R} S_{ijnprt} + U_{ijnt}\right)\right) \\ & + \sum_{i \in V^V \cup V^M \cup V^P}\sum_{n \in PRD_i} HC_{in} \sum_{p=1}^{P}\sum_{r=1}^{R}\sum_{t=1}^{T} I_{ijnprt} \\ & + \sum_{i \in V^C \cup V^D \cup V^R}\sum_{m \in PRD_i} HC_{im} \left(\sum_{t=1}^{T}\left(\sum_{p=1}^{P}\sum_{r=1}^{R} I_{ijmprt} + IU_{imt}\right)\right)\end{aligned}$$

A multi-phase optimization process was used in this study to solve the problem. The first objective function minimizes the total delay cost. The other four remaining objective functions are optimized according to individual considerations. In other words, the four objective functions can be in the sequence (1) Min *SBP*, Min *SBC*, Min *RCP*, and Min *PTI*, or (2) Min *RCP*, Min *SBP*, Min *SBC*, and Min *PTI*, or any other sequence.

The MIP model constructed for this study is extremely complicated. We assumed that there are R demands ($R > 1$ and $DEM_{rp} = D$ for all r and p) with T planning time buckets for P product structures with M items and a supply chain network with V nodes and L links. There are Q substitutions for each item. There are respectively $(V \times M \times P \times R \times T)$ of X_{inprt}, $(V \times M \times T)$ of Z_{int}, $(V \times M \times M \times P \times R \times T)$ of SUB_{imsprt}, $(V \times M \times M \times P \times R \times T)$ of YS_{imsprt}, $(L \times M \times R \times P \times T)$ of S_{ijnprt}, $(L \times M \times T)$ of U_{ijnt}, $(V \times M \times T)$ of IU_{int}, $(M \times R \times P \times T)$ of WL_{iprt}, $(M \times R \times P \times T)$ of WM_{iprt}, $(M \times R \times P \times T)$ of YR_{iprt}, $(R \times P \times T)$ of CN_{rpt}, $(R \times P)$ of PRO_{rp}, and $(V \times M \times P \times R \times T)$ of I_{inprt}. Please note that YR_{iprt}, YS_{imsprt}, CN_{rpt}, and PRO_{rp}, are binary variables. There is a total of $(V \times M \times T)$ constraints (a), (b) and (c), $(R \times P)$ constraint (d), $(V \times M \times T)$ constraints (e), (f) and (g), $(V \times M \times R \times T)$ constraints (h), (i) and (j), $((V + M) \times T)$ constraints (k) and (l), $(V \times M \times P \times T)$ constraints (m), (n) and (o), $(V \times M \times L \times P \times T)$ constraint (p), $(V \times M \times M \times P \times R \times T)$ constraints (q) and (r), $(V \times M \times P \times R \times T)$ constraints (s), (v) and (w), $(V \times M \times L \times P \times R)$ constraint (t), $(V \times M \times L)$ constraint (u), $(V \times M \times L \times T)$ constraint (x), $(V \times M \times T)$ constraints (y) and (z), $(V \times M \times L \times P \times R \times T)$ constraint (aa), $(M \times R \times P \times T)$ constraints (bb), (cc) and (ee), and $(V \times M \times M \times P \times R \times T)$ constraints (dd) and (ff).

References

Ashley, S. (1993). Designing for the environment. *Mechanical Engineering, 115*(3), 52–55.

Awadh, B., Sepehri, N., & Hwaaleshka, O. (1995). A computer-aided process planning model based on genetic algorithms. *Computers & Operations Research, 22*(8), 841–856.

Balakrishnan, A., & Genunes, J. (2000). Requirements planning with substitutions: Exploiting bill-of-materials flexibility in production planning. *Manufacturing & Service Operations Management, 2*(2), 166–185.

Beamon, B. M. (1998). Supply chain design and analysis: Models and methods. *International Journal of Production Economics, 55*(3), 281–294.

Brailsford, S. C., Potts, C. N., & Smith, B. M. (1999). Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research, 119*, 557–581.

Chern, C. C., & Hsieh, J. S. (2007). A heuristic algorithm for master planning that satisfies multiple objectives. *Computers and Operatons Research, 34*(11), 3491–3513.

Chern, C. C., & Huang K. Y. (2007). A heuristic master planning algorithm for green supply chain management. In *Proceedings of the*

38th annual meeting of decision science institute (pp. 361-366), San Francisco.

Chern, C. C., & Yang, I. C. (2006). A heuristic master planning algorithm for supply chains that considers substitutions and commonalities. In *Proceedings of the 37th annual meeting of decision science institute* (pp. 25421–25426), San Antonio, Texas, USA.

Demirtas, E. A., & Ustun, O. (2009). Analytic network process and multi-period goal programming integration in purchasing decisions. *Computer & Industrial Engineering, 56*, 677–690.

Erenguç, S. S., Simpson, N. C., & Vakharia, A. J. (1999). Integrated production/distribution planning in supply chains: An invited review. *European Journal of Operational Research, 115*(2), 219–236.

Evans, G. (1984). An overview of techniques for solving multi-objective mathematical programs. *Management Science, 30*(11), 1268–1282.

Fleischmann, M., Krikke, H. R., & Dekker, R. (2002). Controlling inventory with stochastic item returns: A basic model. *European Journal of Operational Research, 138*, 63–75.

Fleischmann, M., Krikke, H. R., Dekker, R., & Flapper, S. D. P. (2000). A characterization of logistics network for product recovery. *Omega, 28*, 653–666.

Genues, J. (2003). Solving large-scale requirements planning problems with component substitution options. *Computer & Industrial Engineering, 44*, 475–491.

Jayaraman, V., & Pirkul, H. (2001). Planning and coordination of production and distribution facilities for multiple commodities. *European Journal of Operational Research, 133*, 394–408.

Kelbel, J., & Hanzálek, Z. (2011). Solving production scheduling with earliness/tardiness penalties by constraint programming. *Journal of Intelligent Manufacturing, 22*(4), 553–562.

Lakhal, S., Martel, A., Kettani, O., & Oral, M. (2001). On the optimization of supply chain networking decisions. *European Journal of Operational Research, 129*, 259–270.

Lee, Y. H., Jeong, C. S., & Moon, C. (2002). Advanced planning and scheduling with outsourcing in manufacturing supply chain. *Computer & Industrial Engineering, 43*, 351–374.

Lee, Y. H., & Kim, S. H. (2002). Production—distribution planning in supply chain considering capacity constraints. *Computers & Industrial Engineering, 43*, 169–190.

Lyon, P., Milne, R. J., Orzell, R., & Rice, R. (2001). Matching assets with demand in supply-chain management at IBM microelectronics. *Interfaces, 31*(1), 108–124.

Mabini, M. C., & Gelders, L. F. (1991). Repairable item inventory system: A literature review. *Belgian Journal of Operations Research, Statistics and Computer Science, 30*(4), 57–69.

Moon, C., Kim, J., & Hur, S. (2002). Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain. *Computers & Industrial Engineering, 43*, 331–349.

Moon, C., Seo, Y., Yun, Y., & Gen, M. (2006). Adaptive genetic algorithm for advanced planning in manufacturing supply chain. *Journal of Intelligent Manufacturing, 17*(4), 509–522.

Okamoto, A., Gen, M., & Sugawara, M. (2006). Integrated data structure and scheduling approach for manufacturing and transportation using hybrid genetic algorithm. *Journal of Intelligent Manufacturing, 17*(4), 411–421.

Pirkul, H., & Jayaraman, V. (1998). A multi-commodity, multi-plant, capacitated facility location problem: Formulation and efficient heuristic solution. *Computers & Operations Research, 25*(10), 869–878.

Richter, K. (1996). The EOQ repair and waste disposal model with variable setup numbers. *European Journal of Operational Research, 95*(4), 313–324.

Richter, K. (1996). The extended EOQ repair and waste disposal model. *International Journal of Production Economics, 45*(1–3), 443–448.

Sawik, T. (2007). Multi-objective master production scheduling in make-to-order manufacturing. *International Journal of Production Research, 45*(12), 2629–2653.

Schultmann, F., Engels, B., & Rentz, O. (2003). Close-loop supply chains for spent batteries. *Interface, 33*(6), 57–71.

Sheu, J. B., Chou, Y. H., & Hu, C. C. (2005). An integrated logistics operational model for green-supply chain management. *Transportation Research Part E, 41*, 287–313.

Stock, J. R. (1992). *Reverse logistics*. Oak Brook, Illinois: Council of Logistics Management.

Vergara, F. E., Khouja, M., & Michalewicz, Z. (2002). An evolutionary algorithm for optimizing material flow in supply chains. *Computers & Industrial Engineering, 42*, 407–421.