This article was downloaded by: [University of Tennessee, Knoxville]
On: 27 December 2014, At: 14:13
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

International Journal of Computer Integrated Manufacturing
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcim20

Integrated production and distribution planning for single-period inventory products

Yang-Byung Park [a] & Sung-Chul Hong [a]

[a] Department of Industrial Engineering , College of Advanced Engineering, Kyung Hee University , 1 Seocheon-dong, Giheung-ku, Yongin-si, Gyunggi-do, 449-701, Republic of Korea

Published online: 08 May 2009.

To cite this article: Yang-Byung Park & Sung-Chul Hong (2009) Integrated production and distribution planning for single-period inventory products, International Journal of Computer Integrated Manufacturing, 22:5, 443-457, DOI: 10.1080/09511920802527590

To link to this article: http://dx.doi.org/10.1080/09511920802527590

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the “Content”) contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at http://www.tandfonline.com/page/terms-and-conditions

International Journal of Computer Integrated Manufacturing
Vol. 22, No. 5, May 2009, 443–457

Integrated production and distribution planning for single-period inventory products

Yang-Byung Park* and Sung-Chul Hong

Department of Industrial Engineering, College of Advanced Engineering, Kyung Hee University, 1 Seocheon-dong, Giheung-ku, Yongin-si, Gyunggi-do, 449-701, Republic of Korea

(*Received 4 February 2008; final version received 11 September 2008*)

Many firms try separately to optimise their production and distribution functions, but such separation may limit the potential savings. Nowadays, it is more important to analyse these two functions simultaneously by trading off the costs associated with the whole. In this paper, a mixed integer linear programming model is constructed and a hybrid genetic algorithm is proposed incorporating several local optimisation techniques for production and distribution planning problems of single-period inventory products, with the aim of optimally coordinating and integrating the interrelated decisions of production sequencing and vehicle routing. Computational results on the various test problems demonstrate the capability of the proposed algorithm to obtain solutions that are very close to those obtained by the mathematical model for small problems and confirm the effectiveness of the integrated planning approach over the decoupled planning method in which vehicle routing is first developed and a production sequence is subsequently derived. Finally, an investigation is undertaken of the effects of the problem parameters on the effectiveness of the integrated planning approach through sensitivity analysis.

Keywords: integrated and decoupled production and distribution planning; single-period inventory products; hybrid genetic algorithm; supply chain management

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

1. Introduction

Achieving success in the global market requires a fundamental shift in the way business is conducted and affects virtually every aspect of company operations. Recently, a new approach for analysing the supply chain was identified that involves the integration of different functions of the supply chain such as purchasing, production, distribution and warehousing into a single optimisation model. This approach is based on the simultaneous optimisation of decision variables for different functions that have traditionally been optimised separately in order to achieve a synergistic outcome.

However, in practice it is extremely difficult to integrate all the different functions and activities in a supply chain into a single optimisation model. Thus, over the last ten years, most research has focused on a partial integration. The integration of the production and distribution functions of a manufacturing firm, as a step towards full enterprise integration, has gained increasing urgency with the advent of advanced manufacturing and information technologies, the importance of meeting customer needs and the strategic requirement of shortening cycle times (Shapiro 1999, Erenguc *et al.* 2001, Elhedhli and Goffin 2005, Park 2005, Gen and Syarif 2005, Lei *et al.* 2006, Nonino and Panizzolo 2007). Minimising inventory is one strategy used to effectively integrate the production and distribution functions. As inventories are reduced, production becomes more closely linked with distribution.

An extreme example in which production and distribution are inseparable can be found in the production and distribution planning problem for single-period inventory products (PDPSI). Such products are perishable goods that either lose significant value if stored or cause economic loss if delivered late. Examples of single-period inventory products include newspapers, fresh food, and automobile seats. Late delivery of a newspaper or fresh food may result in the loss of a customer, while late delivery of car seats may shut down a production line. The production of single-period inventory products is, in some sense, the limiting example of just-in-time production as it may not be possible to produce perishable goods in advance for inventory. In addition, the total time devoted to both production and distribution for single-period inventory products may be severely limited, thereby further connecting the planning and operation of the two functions.

A little research has been conducted in the area of PDPSI, mostly for daily or weekly publications such as newspapers. Hurter and Buer (1996) developed a

*Corresponding author. Email: ybpark@khu.ac.kr

ISSN 0951-192X print/ISSN 1362-3052 online
© 2009 Taylor & Francis
DOI: 10.1080/09511920802527590
http://www.informaworld.com

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

two-phase heuristic algorithm to determine the production sequence and carrier routes for the newspaper production and distribution problem with setup times between editorial products and strict delivery deadlines. Buer *et al.* (1999) formulated the medium newspaper production and distribution problem into an integrated mathematical model and developed a solution strategy using heuristic search algorithms. Song *et al.* (2002) developed a heuristic to determine the optimal allocation of newspaper agents to printing plants as well as optimal routes for newspaper delivery. They applied the proposed algorithm to a major newspaper company in Korea, reducing the distribution cost by 15% and the delivery time by 40%. Cunha and Mutarelli (2007) proposed a mixed-integer linear programming model to determine the number and location of the magazine manufacturing facilities, an optimal allocation of destinations and the production sequence. They applied their spreadsheet-based optimisation model to a major weekly magazine company in Brazil, yielding a saving of 7.1% in the total cost.

In this paper, the authors construct a mixed integer linear programming model and propose a hybrid genetic algorithm incorporating several local optimisation techniques for PDPSI, with the aim of optimally coordinating the interrelated decisions of the production sequence and vehicle routes for delivery in order to minimise the total cost for production and distribution. An extensive computational study has been performed to evaluate the proposed algorithm by comparing it with the mathematical model and validate the effectiveness of the integrated planning approach by comparing it with the decoupled planning method, which is developed through modifying the Hurter and Buer's (1996) two-phase heuristic algorithm. Finally, the effects of problem size and vehicle capacity on the effectiveness of the integrated planning approach have been investigated through sensitivity analysis.

Many genetic algorithmic approaches to production sequencing have been reported. Aytug *et al.* (2003) and Chaudhry and Luo (2005) have provided a review of genetic algorithms used to solve the production sequencing problem. Considerable research has also been devoted to using genetic algorithms to solve vehicle routing problems. A recent survey on this work is presented by Bräysy and Gendreau (2001, 2005), Cordeau *et al.* (2002), Bräysy *et al.* (2004) and Alvarenga *et al.* (2007).

The remainder of this paper is organised as follows. Section 2 defines PDPSI followed by its mathematical formulation. In section 3, a hybrid genetic algorithm is proposed as an integrated planning solution for PDPSI. An example is solved by applying the proposed algorithm in section 4. In section 5, a computational study is discussed. Finally, section 6 concludes with future research directions.

2. Production and distribution planning problem for single-period inventory products (PDPSI)

PDPSI is considered in the two-echelon supply chain with multi-items (or multi-versions) of a product consisting of a single manufacturing facility and many scattered customers. Timeliness is one of the most important requirements in product delivery. The point of PDPSI is to determine the production sequence of items at a manufacturing facility and the vehicle routes for delivery, which minimise the product-dependent setup, processing, transportation, delay, and vehicle costs.

Two types of PDPSI are considered. In the type I problem, the entire region of customers is divided into several geographic zones based on the item demand. The customers in each zone all order the same single-item. The manufacturing facility produces multi-items and the vehicles deliver the item ordered by the customers in the corresponding zones. An example is the daily newspaper production and distribution problem with several dozen local agents in each geographic zone, where a newspaper printing plant produces several newspaper editions and the vehicles deliver the newspaper edition to local agents in the corresponding zone. In the type II problem, customers dispersed over a whole region order a single- or multi-item. The manufacturing facility produces multi-items and the vehicles deliver the items ordered by the customers. An example is the bakery production (doughnuts) and delivery problem with several hundred retailers in one city.

Customer order numbers are assigned to the customers' item orders in order of the item number ordered by the customers, starting with customer 1. For example, let us suppose that customer 1 orders items 1 and 2, and customer 2 orders item 2. Then, customer 1's order of item 1 is assigned customer order number 1, customer 1's order of item 2 is assigned customer order number 2, and customer 2's order of item 2 is assigned customer order number 3. When customers order a single item, the customer order number is equal to the customer number.

The manufacturing facility produces multi-items of a product sequentially in a single production line. The production quantity of each item is determined by its total demand from customers in a single period, with no provisions for shortages. The setup time between product items depends on the previously produced item. Owing to the dissimilarity of items, the setup effort required to shift between products is an issue.

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

Loading is started after all items to be loaded are produced. Mixed loading of different items is allowed. Loading and unloading times are proportional to the work quantity. A split delivery of order quantity is not allowed, but a split delivery of items is allowed when a customer orders multi-items. In both types of problems, there are common soft and hard deadlines for customers. When the delivery is delayed beyond the soft deadline, the delay cost is proportional to the delivery quantity and time delay. The hard deadline must be met.

A formulation of PDPSI is as follows:

$$\text{Minimise } Z = C_1 \sum_{h=1}^{M} \left(p_h \sum_{j=1}^{L} q_{hj} + \sum_{\substack{g=0 \\ g \neq h}}^{M} s_{gh} y_{gh} \right) + C_2 \sum_{i=0}^{L} \sum_{\substack{j=0 \\ j \neq i}}^{L} \sum_{v=1}^{R} t_{ij} x_{ijv} + C_3 \sum_{h=1}^{M} \sum_{j=1}^{L} q_{hj} l_j + C_4 \sum_{j=1}^{L} \sum_{v=1}^{R} x_{0jv} \tag{1}$$

s.t.

$$\sum_{\substack{g=0 \\ g \neq h}}^{M} y_{gh} = 1, \quad h = 0, \dots, M \tag{2}$$

$$\sum_{\substack{h=0 \\ h \neq g}}^{M} y_{gh} = 1, \quad g = 0, \dots, M \tag{3}$$

$$m_g - m_h + M y_{gh} \leq M - 1, \quad g = 1, \dots, M; \; h = 1, \dots, M; g \neq h \tag{4}$$

$$z_h - z_g - B y_{gh} \geq s_{gh} + p_g \sum_{j=1}^{L} q_{hj} - B, \quad g = 0, \dots, M; h = 0, \dots, M; g \neq h \tag{5}$$

$$\sum_{j=1}^{L} q_{hj} \sum_{\substack{i=0 \\ i \neq j}}^{L} x_{ijv} \leq B w_{hv}, \quad h = 1, \dots, M; v = 1, \dots, R \tag{6}$$

$$a_j \geq z_h + B w_{hv} + b \sum_{n=1}^{L} q_{hn} \sum_{\substack{i=0 \\ i \neq n}}^{L} x_{inv} + t_{0j} x_{0jv} - B, \quad j = 1, \dots, L; h = 1, \dots, M; v = 1, \dots, R \tag{7}$$

$$a_j - a_i \geq B x_{ijv} + u_i \sum_{h=1}^{M} q_{hi} + t_{ij} - B, \quad i = 1, \dots, L; j = 1, \dots, L; v = 1, \dots, R; i \neq j \tag{8}$$

$$a_j \leq E, \quad j = 1, \dots, L \tag{9}$$

$$a_j + k_j - l_j = D, \quad j = 1, \dots, L \tag{10}$$

$$\sum_{\substack{j=0 \\ j \neq i}}^{L} x_{ijv} \leq 1, \quad i = 0, \dots, L; v = 1, \dots, R \tag{11}$$

$$\sum_{\substack{j=0 \\ j \neq i}}^{L} x_{ijv} - \sum_{\substack{j=0 \\ l \neq i}}^{L} x_{jiv} = 0, \quad i = 0, \dots, L; v = 1, \dots, R \tag{12}$$

$$f_{ijv} - Q x_{ijv} \leq 0, \quad i = 0, \dots, L; j = 0, \dots, L; v = 1, \dots, R; i \neq j \tag{13}$$

$$\sum_{\substack{j=0 \\ j \neq i}}^{L} f_{ijv} - \sum_{\substack{j=0 \\ l \neq i}}^{L} f_{jiv} \leq 0, \quad i = 1, \dots, L; v = 1, \dots, R \tag{14}$$

$$\sum_{v=1}^{R} \left(\sum_{\substack{j=0 \\ j \neq i}}^{L} f_{jiv} - \sum_{\substack{j=0 \\ l \neq i}}^{L} f_{ijv} \right) = \sum_{h=1}^{M} q_{hi}, \quad i = 1, \dots, L \tag{15}$$

$$x_{ijv} = \{0,1\}, y_{gh} = \{0,1\}, w_{gv} = \{0,1\}, m_h \geq 1, f_{ijv} \geq 0, z_{gh} \geq 0, b_{gh} \geq 0, a_j \geq 0, l_j \geq 0, k_j \geq 0, z_0 = 0, p_0 = 0, \forall i,j,v,g,h \tag{16}$$

where

- C_1: production cost conversion factor,
- C_2: transportation cost conversion factor,
- C_3: delay cost conversion factor,
- C_4: vehicle cost conversion factor,
- 0: manufacturing facility (depot),
- N: number of customers,
- M: number of items,
- L: largest customer order number ($L = N$ in type I problems),
- R: number of vehicles,
- Q: vehicle capacity,
- D: common soft deadline,
- E: common hard deadline,
- B: very large number,
- p_h: processing time of item h,
- s_{gh}: setup time of item h when g is produced immediately prior (g = 0 means that h is the first item in the production sequence),
- t_{ij}: travel time from location of customer order number i to one of customer order number j,
- b: unit loading time at a manufacturing facility,
- u_i: unit unloading time at location of customer order number i
- q_{hi}: order quantity of item h of customer order number i,

a_j: arrival time at location of customer order number j,
f_{jv}: loaded quantity when vehicle v arrives at location of customer order number j,
m_h: order of item h in the production sequence,
z_h: production finish time of item h,
l_j: delay time of customer order number j,
w_{hv}: 1 if item h is loaded on vehicle v; 0 otherwise,
x_{ijv}: 1 if vehicle v travels from location of customer order number i to one of customer order number j; 0 otherwise,
y_{gh}: 1 if item h is produced right after g; 0 otherwise ($g = 0$ means that h is the first item in the production sequence).

The objective function (1) seeks to minimise the total of the product-dependent setup, processing, transportation, delay, and vehicle costs. Equations (2) and (3) determine the production sequence of items. Equation (4) eliminates the subtours of production sequence. Equation (5) determines the production finish times of items. Equations (6) and (7) compute the arrival times at the first route locations. Equation (8) computes the arrival times at the route locations after the first ones. Equations (9) and (10) are the constraints for delivery deadlines. Equations (11) to (15) are for vehicle routing (Abdelmaguid and Dessouky 2006).

3. Hybrid genetic algorithm for PDPSI (HGAP)

A hybrid genetic algorithm is proposed to solve PDPSI, incorporating a local optimisation as an add-on extra to the genetic algorithm loop of recombination and selection. The proposed algorithm termed HGAP (Hybrid Genetic Algorithm for PDPSI), guarantees the generation of feasible solutions in its whole evolutionary process by using the special representation scheme of a solution. HGAP employs a repair process to correct for inefficient routes with multiple visits to the same customer. The multiple visits are merged into a single visit through this repair operation. HGAP adopts an elitist replacement strategy to keep track of the best solution in the evolution process. The procedure of HGAP is described as follows:

Step 1: Construct the initial population of n chromosomes.

Step 2: Conduct crossover and mutations to the chromosomes selected on the basis of their rates in the population and perform the local optimisation on the production sequence and vehicle routes of the offspring generated.

Step 3: If a termination condition is satisfied, report the best chromosome found so far and stop. Otherwise, construct the new population consisting of the m best chromosomes in the current population and the n–m best offspring obtained in Step 2, and return to Step 2.

3.1. Solution coding

A solution code is a mixed scheme of binary and permutation representation. A chromosome consists of three parts: a local optimisation algorithm to apply, production sequence, and vehicle routes.

The first part contains six genes with binary digits, whose position corresponds to the local optimisation algorithm number: the first two genes are for the production sequence and the remaining four for the vehicle routes. A gene with '1' denotes the application of the corresponding local optimisation algorithm. Only one '1' is assigned to the first two genes and the remaining four genes, respectively. The other four genes are assigned '0'.

The second part represents the permutation of item numbers, that is, the production sequence of items. Therefore, the number of genes in the second part is equal to the number of items. The production start time of an item is determined by adding its setup time to the sum of setup times and processing times for the precedent items in the production sequence.

The third part represents the vehicle routes using customer order numbers. The route sequence follows the customer numbers corresponding to the customer order numbers listed in the third part (left-to-right scan procedure). A vehicle route always starts with depot 0. The vehicle start time is computed by summing the production finish time of the items to be loaded and their loading time.

When there is more than one item order for the same customer apart, the route is repaired by moving the order number(s) of that customer to the position right after the order number for the same customer at the leftmost position, in order to avoid multiple visits of a vehicle to the same customer during the delivery tour.

For M items and R routes, a solution coding is as follows:

$$S = (\underbrace{l_1..l_i.l_6}_{1^{\text{st}}\text{ part}}\ \underbrace{p_{[1]} \ldots p_{[i]} \ldots p_{[M]}}_{2^{\text{nd}}\text{ part}}\ \underbrace{0x_{[1]1} \ldots x_{[i]1} \ldots x_{[n_1]1}0 \ldots\ldots 0x_{[1]v} \ldots x_{[i]v} \ldots x_{[n_v]v}0 \ldots\ldots 0x_{[1]R} \ldots x_{[i]R} \ldots x_{[n_R]R}}_{3^{\text{rd}}\text{ part}})$$

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

where

- l_i: 1 if the local optimisation algorithm #i is applied; 0 otherwise,
- $p_{[i]}$: ith item number in the production sequence,
- $x_{[i]v}$: ith customer order number in the sequence of route v, $1 \leq v \leq R$,
- n_v: the number of customers' item orders in route v.

For example, using the customer order numbers in the previous example, $S = (1001002101302)$ is decoded as follows: Local optimisation algorithms #1 and #4 are applied to the production sequence and vehicle routes, respectively. The production sequence is 2–1. Two vehicles are required, whose routes are 0-customer 1 (item 1)-customer 2 (item 2) $-$ 0 and 0-customer 1 (item 2) $-$ 0. The item number recorded inside the parentheses denotes the one ordered for that customer.

3.2. *Initial population*

For the initial population, 90% of the chromosomes are randomly generated, and the rest are generated by applying heuristics to the second and third parts of the chromosome. This strategy is referred to as a hybrid population initialisation. In their empirical study of genetic algorithmic approaches, Schmitt and Amini (1998) observed that a hybrid population initialisation produces the best results for all TSP type problems with respect to solution quality and search time.

In order to create random initial chromosomes, the first part of a chromosome is constructed by assigning '1' randomly to one of the first two genes and one of the next four genes, respectively. The remaining four genes are assigned '0'. Genes of the second and third parts of a chromosome are determined by a random generation of the item number permutation and customer order number permutation, respectively. Finally, '0' is inserted right before the first customer order number of each route in the third part.

In order to create the initial chromosomes by applying heuristics, the first part of a chromosome is randomly constructed using the same method as used for the random chromosomes. The second part of a chromosome is constructed by applying a stochastic insertion heuristic. The heuristic starts with constructing an initial partial production sequence by placing the item selected randomly at the first position. The item to be placed at the next position is determined on the basis of the selection probability of the rth item, $P(r) = k_1 k_2^{r-1} (0 < k_1 < 1, 0 < k_2 < 1)$, where r is the rank of the non-inclusive items in the current partial production sequence arranged in ascending order of their setup times. This process is repeated until the complete production sequence is formed in the second part. The selection probability distribution is designed such that the item requiring less time for setup change obtains a higher probability of selection for the next item to produce.

The third part of a chromosome is constructed by applying the stochastic approach of the Clarke and Wright (1964) savings algorithm. The heuristic algorithm begins by designating a separate vehicle for each customer order number. A pair of customer order numbers that are to be linked is selected from the savings list on the basis of the selection probability of the rth pair, $P(r) = k_1 k_2^{r-1} (0 < k_1 < 1, 0 < k_2 < 1)$, where r is the rank of feasible pairs in the savings list arranged in descending order of savings. The feasible pair guarantees that the restriction of vehicle capacity and the hard deadline are kept in the newly formed route. This process is repeated until no further route connection is feasible. Finally, '0' is inserted right before the first customer order number of each route in the third part.

3.3. *Fitness evaluation and selection*

For the selection process, a roulette wheel method is applied using the relative fitness of a chromosome. The relative fitness of chromosome k is calculated as follows:

$$f_k = \frac{z_{\max} - z_k}{z_{\max} - z_{\min}} \tag{17}$$

where $z_{\max}$ and $z_{\min}$ denote the maximum and minimum total costs of the chromosomes in the population, respectively. The total cost of a chromosome is computed by using the objective function of PDPSI. Thus, $0 \leq f_k \leq 1$.

3.4. *Crossover*

Crossover is conducted separately for the three parts of the chromosome selected according to the crossover rate p_c. Two offspring are generated from two parents by the crossover operator.

For the first part of a chromosome, BX (biased crossover) is applied to the first two genes and the next four genes, respectively. BX is conducted on the basis of the parent's improvement rate, which is a reduction percentage of the parent's total cost by the local optimisation in the previous generation. To build the first two genes of an offspring, a parent is selected on the basis of the probability in proportion to the improvement rates of two parents obtained by the local optimisation for the production sequence. Then, bits of the first two genes of the selected parent are

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

copied together to the corresponding genes of offspring. The rationale behind this crossover method is to increase the possibility of re-applying the local optimisation algorithm that showed a better performance in the previous evolution phase. If the improvement rates of two parents are equally zero, then a gene of '1' is randomly determined in two offspring, which in effect introduces a mutation. The same procedure is applied to build the next four genes of offspring, employing the improvement rates of two parents obtained by the local optimisation for the vehicle routes.

For example, for two parents, $P_1 = (100010\ldots)$ and $P_2 = (011000\ldots)$, the improvement rates based on the local optimisation for production sequence are $I_1 = 0.05$ and $I_2 = 0.15$. The two random numbers generated from the range [0..1] are $R_1 = 0.55$ and $R_2 = 0.81$. The improvement rates for the two parents based on the local optimisation for vehicle routes are $I_1 = 0.08$ and $I_2 = 0.02$. The two random numbers generated are $R_1 = 0.21$ and $R_2 = 0.96$. Then, the selection probabilities of the two parents are computed as $S_1 = 0.25$ (=0.05/(0.05 + 0.15)) and $S_2 = 0.75$ (=0.15/(0.05 + 0.15)) for the first two genes of the offspring and $S_1 = 0.8$ (=0.08/(0.08 + 0.02)) and $S_2 = 0.2$ (=0.02/(0.08 + 0.02)) for the next four genes of the offspring. Thus, two offspring are obtained as $O_1 = (010010\ldots)$ and $O_2 = (011000\ldots)$.

For the second and third parts of a chromosome, PMX (partially mapped crossover) proposed by Goldberg and Lingle (1985) is conducted to one of the two parts. PMX is a two-cut point crossover method with mappings, originally developed for TSP. The part for conducting PMX is selected on the basis of the probability proportional to the number of genes in each of the second and third parts. PMX transmits good schemata to the offspring and simultaneously exploits important similarities in the value and ordering. This property is validated through the computational experiments in Section 5. When conducting crossover on the third part, all '0' values should be removed in the third part of the two parents before the operation. After the crossover operation, '0' is inserted right before the first customer order number for each route in the two offspring. When necessary, the repair process is performed on the routes of the offspring in order to correct for inefficient routes.

3.5. *Mutation*

As the third part of a chromosome represents multiple vehicle routes, the classical mutation may cause random breaking, sub-tours, an unnecessary increase in fleet size, and repetition in customer order numbers, which, in combination, can result in a loss of the optimality information gained by the genetic algorithm over the previous generations. Furthermore, performing local optimisation may restore offspring to the chromosome before mutation. Therefore, two new mutation methods are proposed: random and merge. Random mutation substitutes a chromosome with a new randomly generated one. Merge mutation attempts to combine multiple visits to the same customer separated by different routes into a single visit. The repair process may be required after mutation to correct for inefficient routes.

The number of chromosomes for random mutation in each generation varies. The expected number is computed by multiplying the population size by the adapted random mutation rate, p_r. In each generation, the adaptive random mutation rate is determined by

$$p_r = p_r^0 (z_{\min} / z_{\text{ave}}) \tag{18}$$

where p_r^0 is the initial value of p_r, and $z_{\min}$ and z_{ave} are the minimum and average total costs of chromosomes in the population, respectively. If the population diversity becomes very small, $z_{\min}/z_{\text{ave}}$ is close to 1.0 and therefore p_r is reset to p_r^0, which is the highest allowable value for the random mutation rate. On the other hand, if the population diversity becomes very large, $z_{\min}/z_{\text{ave}}$ is close to 0.0 and the number of chromosomes for random mutation is thus very small. This strategy of adaptive mutation prevents the genetic search from a premature convergence and guarantees effectiveness.

Merge mutation randomly selects one among the same customer orders separated by different routes and moves the others right after it as long as the route feasibility condition remains satisfied with respect to the vehicle capacity and hard deadline. If orders from the same customer are not separated into different routes in the chromosome, merge mutation is not applied. The chromosomes for merge mutation are selected according to the merge mutation rate, p_g. Merge mutation may reduce the vehicle travel time and delay by delivering as many items as possible for the same customer in a single visit.

3.6. *Local optimisation*

A total of six greedy algorithms (Cordeau *et al*. 2002, Gottlieb *et al*. 2003) are considered for the local optimisation. Among them, two are for the production sequence and four are for the vehicle routes. The six greedy algorithms are as follows: an or-opt algorithm for the production sequence (#1), a 2-opt exchange algorithm for the production sequence (#2), an or-opt algorithm for each of all vehicle routes (#3), a 2-opt exchange algorithm for each of all vehicle routes (#4), an or-opt algorithm for all the adjacent routes (#5), and a 2-opt exchange algorithm for all the adjacent routes (#6). Based on the positions of the two '1's in the first part of a chromosome, the corresponding local optimisation algorithms are applied to both the production sequence and vehicle routes, one after another. The local optimisation of the production sequence affects the route feasibility with respect to the hard deadline. Therefore, the route feasibility must be checked whenever the production sequence is changed.

4. Example

An example problem of type II with $N = 50$ and $M = 5$ is solved using HGAP. Among 50 customers, 10 order three items, 15 order two items, and 25 order a single-item. So, $L = 85$. The manufacturing facility is located at (40, 40). The coordinates of customer locations are generated from a uniform distribution of [5, 85]. The order quantity of an item from a customer is generated from a uniform distribution of [10, 43]. The capacity of all vehicles is set to 500. For all customers, the hard and soft deadlines are set to 160 and 128 (or 80% of the hard deadline), respectively. Production times for items per unit, initial production setup times, and product-dependent setup times are generated from a uniform distribution of [0.03, 0.04], [10, 15], and [2, 9], respectively. Loading and unloading times per unit are equally set to 0.02. Travel times between two locations are determined by half of the Euclidean distance between them. In the objective function, $C_1 = 10$, $C_2 = 1$, $C_3 = 0.01$, and $C_4 = 50$. The population size is set to 60. The two parameters for creating the initial chromosomes are set as $k_1 = 0.6$ and $k_2 = 0.4$. For the crossover and mutation operations, $p_c = 0.75, p_r^0 = 0.05$, and $p_g = 0.03$.

HGAP is programmed in Visual Basic 6.0, and the example is solved on an IBM compatible Pentium IV PC with 256 RAM and 1.4 GHz. The convergence behaviour of HGAP in the generations for the example problem is shown in Figure 1. It is seen that the solution is gradually converged on after the active improvement phase of the first 500 generations. HGAP had a long computation time, about 30 CPU minutes for 1000 generations.

The production sequence of the solution obtained by HGAP is C-E-A-B-D. Table 1 presents the production sequence, production quantity, setup time, production start and finish times, and setup and processing costs for the five items. The production cost is defined as the sum of the setup and processing costs. Table 2 presents the route, departure time from the manufacturing facility, arrival time at the last customer, and transportation, delay, vehicle costs for the seven vehicles required. The distribution cost is defined as the sum of transportation, delay, and vehicle costs. The routes of the seven vehicles are shown in Figure 2. The lines connecting the depot and customers are omitted for visibility.

It is seen from Table 2 that the delivery items for the seven vehicles are ABDE, ACDE, ABCDE, ABDE, ABCE, ABCDE and BCE. Because vehicles 5 and 7 do not deliver item D, which is the last one to be produced, they can leave the depot earlier than the other vehicles and visit more customers over a longer travel time, thereby decreasing the delay cost but increasing the transportation cost. On the other hand, vehicle 6, which leaves the depot at the latest time

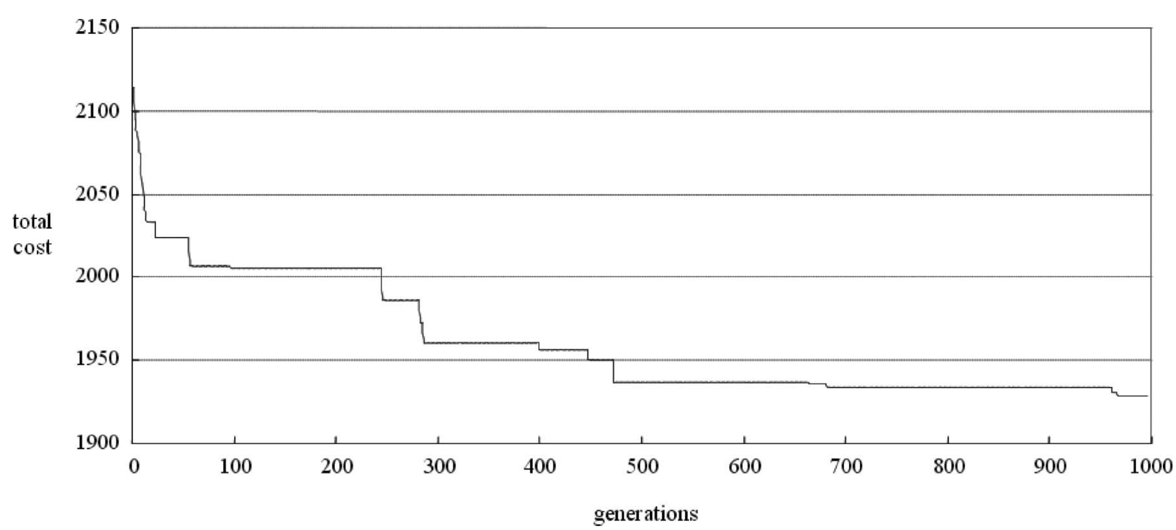

Figure 1. Total cost over generations for the example problem.

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

Table 1. Production sequence and related information of the solution obtained by HGAP for the example problem.

Production sequence	Product item	Production quantity	Setup time	Production start time	Production finish time	Setup cost	Processing cost	Production cost
1	C	494	11	11.0	28.0	110	170	280
2	E	510	2	30.0	47.3	20	173	193
3	A	459	2	49.3	64.5	20	152	172
4	B	406	4	68.5	82.5	40	140	180
5	D	316	7	89.5	100.7	70	112	182
Sum		2185	26			260	747	1007

Table 2. Vehicle routes and related information of the solution obtained by HGAP for the example problem.

Route no.	Vehicle routes (Delivery items)	Departure time from the manufacturing facility	Arrival time at the last customer	Transportation cost	Delay cost	Vehicle cost	Distribution cost
1	0-2(A,B)-18(A)-35(D)-3(A,D)-23(E)-49(B,E)-**6**(D,E)-**4**(D)-0	106.4	156.4	54.6	30.8	50	135.4
2	0-19(D)-47(C,D,E)-46(C)-12(D,E)-5(E)-16(A,C,D)-0	106.3	155.4	66.1	30.8	50	146.9
3	0-32(B)-9(C)-**4**(B)-17(A,E)-**6**(C)-38(D)-0	104.2	152.0	55.9	4.6	50	110.5
4	0-14(A,E)-13(A)-37(D,E)-21(A,E)-28(A,D,E)-42(B)-0	106.5	156.1	56.7	24.4	50	131.1
5	0-50(A)-26(A)-45(B,E)-33(A,B)-40(A)-41(A,C)-24(C)-44(A,B,C)-1(E)-20(E)-36(B)-0	90.5	152.5	64.0	16.3	50	130.3
6	0-25(B,C,E)-15(D)-30(A,C,D)-31(A,C,E)-27(A,E)-**11**(A,B)-**22**(E)-0	108.2	155.0	52.2	39.4	50	141.6
7	0-39(E)-**22**(B,C)-**11**(C)-48(B,C)-43(B)-34(B)-7(B,C)-10(B)-29(C)-8(C)-0	89.8	142.9	64.0	12.8	50	126.8
Sum				413.5	159.1	350	922.6

after loading all five items, generates the highest delay cost. The customers with a split delivery of items by different vehicles are boldly expressed in each vehicle route. For example, items B and D for customer 4 are separately delivered by vehicles 3 and 1, respectively. The split delivery to customers 4 and 22 occurs because of the vehicle capacity, and the split delivery to customers 6 and 11 is due to the production sequence.

5. Computational study

5.1. Decoupled planning algorithm for PDPSI

In order to mimic a decoupled approach to PDPSI commonly found in industry, the Hurter and Buer (1996)'s two-phase heuristic algorithm has been modified, which was developed for the newspaper production and delivery problem with different editions in demographic regions. The modified algorithm, named PH&B, treats the production sequencing and vehicle routing separately and does not allow the split delivery of items ordered by the same customer. The procedure of PH&B is summarised as follows:

Step 1: The route construction begins with connecting the customer farthest from the manufacturing facility and then continuously connecting the nearest customer to the one connected immediately before until the vehicle capacity is met. The route construction process is repeated with the remaining customers until all customers are routed. In the type I problem, the route is formed within a zone and contains only a single delivery item.

Step 2: Compute the arrival time for the last customer in each route, assuming that all items are produced at the manufacturing facility at time 0.

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

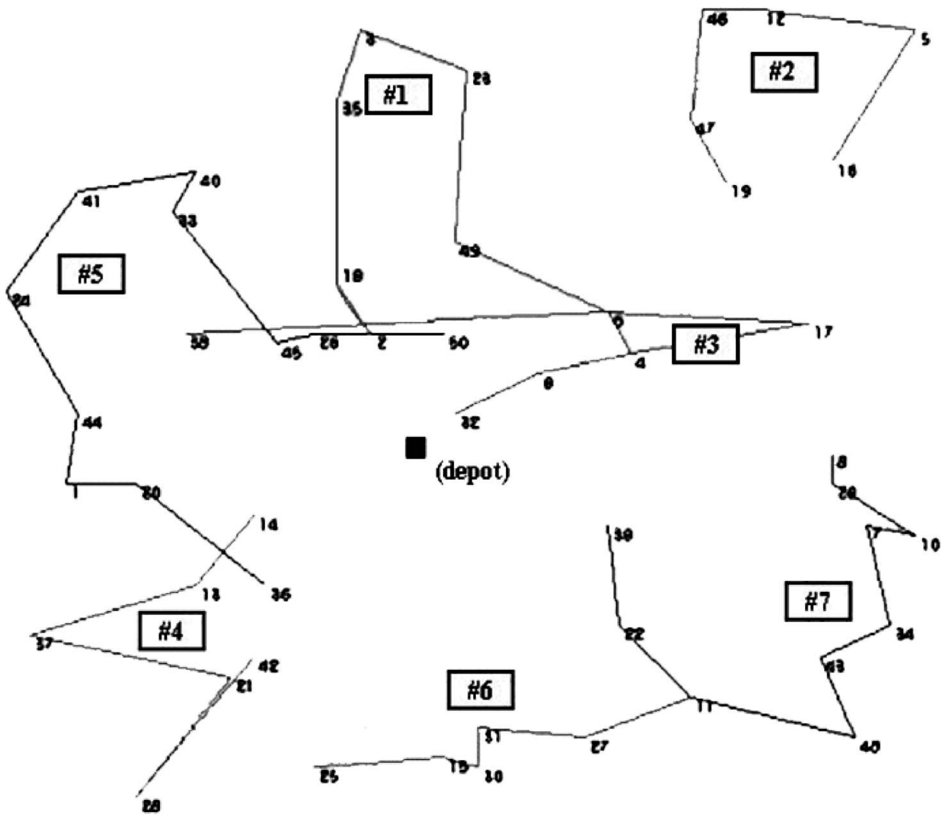

Figure 2. Vehicle routes of the solution obtained by HGAP for the example problem.

Arrange the routes in descending order of arrival time to the last customer. The arrival time at the last customer on the route is computed by summing the loading time, travel time, and unloading times for customers prior to the last customer. Arbitrarily select one delivery item from the first ordered route and assign it to the first position in the production sequence.

Step 3: If there are items in the route, which have not been positioned in the current partial production sequence, select the item with the shortest setup time and assign it to the last position of the current partial production sequence. Otherwise, proceed to Step 4. Continue this assignment process until no more unassigned items are left in the route.

Step 4: Select the next ordered route and repeat Step 3. When the production sequence is completed, move to Step 5.

Step 5: For each route, check the customers with respect to the hard deadline by computing the exact arrival times for each based on the production sequence obtained in Step 4. If there are infeasible routes, remove all the customers who violate the hard deadline from the routes and start Step 1 with the removed customers. Otherwise, stop.

5.2. *Comparison of HGAP and the mathematical model*

HGAP has been evaluated by comparing its performance with the mathematical model on 9 type I and 9 type II small test problems. The comparison was limited to small problems because of the tremendous computation required by the mathematical models to obtain the optimal solutions to larger problems. The mathematical models were solved using CPLEX 9.1 (ILOG 2005). All test problems were solved on an IBM compatible Pentium IV PC with 256 RAM and 1.4 GHz.

In type I test problems, $N = 15, 18, 21$ and $M = 6$. In type II problems, $N = 8, 10, 12$ and $M = 6$. The manufacturing facility is located at (40, 40). The coordinates of customer locations are generated according to a uniform distribution of [5, 85]. In type I problems, a whole customer area of 90*90 is divided into six equal size squares and a different item is assigned to each square. In type II problems, among customers, 20% order three items, 30% order two

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

items, and 50% order a single item. Therefore, for example, $L = 17$ when $N = 10$ in the type II problem.

The order quantity of an item from a customer is generated from a uniform distribution of [10, 43]. Soft and hard deadlines for customers are set to 120 and 150, respectively. The capacity of all vehicles is equally set to 300. Unit processing times of items, initial production setup times, and product-dependent setup times are drawn according to a uniform distribution of [0.06, 0.08], [10, 15] and [2, 9], respectively. Loading and unloading times per unit are uniformly set to 0.02. Travel times between two locations are determined by half of the Euclidean distance between them.

In the objective function, $C_1 = 10$, $C_2 = 1$, $C_3 = 0.01$, and $C_4 = 50$. The population size is set to 60. The two parameters for creating the initial chromosomes are set to $k_1 = 0.6$ and $k_2 = 0.4$. For the crossover operation, $p_c = 0.75$ in both types of problems. For the mutation operation, $p_r^0 = 0.08$ and $p_g = 0.0$ in type I problems, and $p_r^0 = 0.05$ and $p_g = 0.03$ in type II problems. p_g is set to zero in type I problems because merge mutation is not needed. The evolution of HGAP is stopped when the total number of generations reaches 1000. The cost conversion factors and genetic parameter values were determined through a pilot study on HGAP using several larger test problems, while considering the balance of production and distribution costs, solution quality, and computation time.

Tables 3 and 4 compare HGAP with the mathematical model with respect to costs and computation time on type I and II problems, respectively. The error rate of HGAP to the solution of the mathematical model, computed by the equation: 100*(total cost obtained by HGAP - total cost obtained by the mathematical model)/total cost obtained by the mathematical model, remains very low; 0.15% and 0.21% on average in type I and II problems, respectively. HGAP found the same solutions as the mathematical model in 6 of 18 problems. The error rates would be expected to be higher on larger problems. However, the magnitude of error may be acceptable to obtain a good solution quickly for the complex planning problems that cannot be easily solved in an optimal way.

5.3. *Comparison of HGAP and PH&B*

The integrated planning approach has been evaluated by comparing the performance of HGAP with PH&B on ten type I and II test problems, each. PH&B is programmed in Visual Basic 6.0. In all the test problems, $N = 100$ and $M = 9$. A whole customer area of 90 × 90 is divided into nine equally sized squares. The order quantity of an item from a customer is generated from a uniform distribution of [30, 60] and [10, 43] in type I and II problems, respectively. Thus, the total order quantity from a customer becomes 45 on average in both type I and II problems. For type II problems, the total order quantity from a customer on average is computed by $[(20 \times 3 + 10 \times 2 + 50 \times 1)/100] \times (10 + 43)/2$. In both types of problems, soft and hard deadlines for customers are set to 200 and 250, respectively. The capacity of all vehicles is equally set to 500. Unit processing times of items are drawn according to a uniform distribution of [0.03, 0.04]. The evolution of HGAP is stopped when the total number of generations reaches 200. Other problem input data and genetic parameters are the same as in section 5.2.

Table 5 compares HGAP and PH&B with respect to the costs and computation time on type I problems. It is clear that HAGP always obtains better solutions than PH&B, with an average total cost reduction rate of 15.8%. The total cost reduction rate is computed by the equation: (total cost obtained by PH&B-total cost obtained by HGAP)/total cost obtained by PH&B. The cost reduction with HGAP is attained for all cost elements except for the delay cost in problems 7 and 9. This outcome occurs because HGAP constructs some

Table 3. Comparison of HGAP and the mathematical model on type I problems.

		HGAP				Mathematical model				
Problem	$N(L)$	Production cost	Distribution cost	Total cost	CPU time (sec)	Production cost	Distribution cost	Total cost	CPU time (sec)	Error rate (%)
1	15(15)	501.8	270.7	772.5	284	501.8	270.7	772.5	2415	0.00
2	15(15)	560.4	274.1	834.5	275	560.4	273.0	833.4	508	0.13
3	15(15)	565.1	272.5	837.6	243	565.1	271.8	836.9	2657	0.08
4	18(18)	595.4	290.7	886.1	281	595.4	289.2	884.6	3278	0.17
5	18(18)	530.9	280.0	810.9	311	530.9	280.0	810.9	5441	0.00
6	18(18)	530.3	277.2	807.5	267	530.3	276.1	806.4	4705	0.14
7	21(21)	569.8	301.2	871.0	573	569.8	298.7	868.5	2417	0.29
8	21(21)	594.3	306.5	900.8	356	594.3	306.5	900.8	10791	0.00
9	21(21)	586.7	301.5	888.2	560	586.6	297.0	883.6	7523	0.52

Table 4. Comparison of HGAP and the mathematical model on type II problems.

		HGAP				Mathematical model				
Problem	$N(L)$	Production cost	Distribution cost	Total cost	CPU time (sec)	Production cost	Distribution cost	Total cost	CPU time (sec)	Error rate (%)
1	8(13)	459.3	239.0	698.3	549	459.3	237.5	696.8	12623	0.22
2	8(13)	447.0	232.7	679.7	395	447.0	232.7	679.7	9010	0.00
3	8(13)	480.5	238.8	719.3	403	480.5	237.0	717.5	9540	0.25
4	10(17)	534.3	248.6	782.9	496	534.3	247.0	781.3	15470	0.20
5	10(17)	524.7	250.0	774.7	485	524.7	247.2	771.9	14036	0.36
6	10(17)	526.9	248.6	775.5	441	526.9	248.6	775.5	17541	0.00
7	12(20)	593.4	257.7	851.1	504	593.4	251.9	845.3	24053	0.69
8	12(20)	602.0	266.2	868.2	585	602.0	264.6	866.6	15470	0.18
9	12(20)	609.0	268.3	877.3	500	609.0	268.3	877.3	27443	0.00

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

routes by including the customers in several adjacent zones in the two problems in order to reduce the transportation and vehicle costs. As a result, the vehicle departure is delayed at the manufacturing facility until all the items to be loaded have been produced. The highest reduction rate of total cost with HGAP is 22.7% in problem 10. In that problem, most customers are clustered near the border of the adjacent zones, so HGAP is able to reduce the distribution cost significantly by including as many customers near the border of the adjacent zone as possible with a small increase in the vehicle travel distance. As shown in Table 5, the computation time for HGAP is much longer than that for PH&B.

Table 6 compares HGAP and PH&B with respect to the costs and computation time on type II problems. HGAP always obtains better solutions than PH&B, with an average total cost reduction rate of 23.3%. The cost reduction with HGAP is attained in all cost elements, especially in the delay cost. In type II problems, about 52.4% of the total cost reduction with HGAP is achieved through delay cost savings, which is over 10 times greater than the reduction in type I problems. The much higher delay cost by PH&B in type II problems is due to the characteristics of type II problems and the routing procedure of PH&B. That is, in type II problems, PH&B constructs the routes by connecting the nearest customers with multi-item demand dispersed over the whole geographical region, solely in order to reduce the travel distance. As a result, the route constructed contains almost all items for delivery and so the vehicle departure is delayed at the manufacturing facility until all the items to be loaded are produced, thereby significantly increasing the number of delayed customers. This is the critical drawback resulting from the decoupled planning of production and distribution. However, HGAP searches for the best solutions by simultaneously reducing the production and distribution costs in an integrated way during its evolution process.

When comparing the performance of HGAP for two problem types, HGAP shows about a 1.5 times higher total cost reduction rate in type II problems than in type I problems, confirming the greater effectiveness of HGAP in type II problems. Moreover, HGAP has a much longer computation time in type II problems than in type I problems. This is because the number of item orders from customers in type II problems is 1.7 times more than that in type I problems.

Through the comparison of HGAP and PH&B on two types of PDPSI, it has been shown that the integrated planning approach is superior to the decoupled planning approach with respect to the solution quality. Three major reasons for the superiority of the integrated planning approach are summarised as follows: (i) HGAP searches for the best route solution with the actual start time from the manufacturing facility computed on the basis of the production sequence, that is, the production sequencing and vehicle routing are treated in an integrated way; (ii) HGAP selectively applies the local optimisation algorithms to both the production sequence and vehicle routes of the offspring generated on the basis of their performance in the previous generation; and (iii) HGAP attempts the vehicle consolidation and split delivery of items in its evolution process, while considering the increase in the production cost.

An evaluation of the effect of the problem parameters on the effectiveness of HGAP over PH&B was undertaken by performing a sensitivity analysis on problem size and vehicle capacity with respect to the total cost reduction rate and computation time of HGAP for type II problems. The problem size is determined by the number of customers. The vehicle cost conversion factor is set by using the regression equation, $C_4 = 0.075Q + 12.5$, which is derived from actual data on vehicle price for capacity. The hard deadline is set by increasing it by 20 as the vehicle capacity is increased by 100, starting from $E = 210$ for $Q = 300$. The soft deadline is set to 80%

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

Table 5. Comparison of HGAP and PH&B on type I problems.

Problem	HGAP						PH&B						Total cost reduction rate (%)
	Production cost	Transportation cost	Delay cost	Vehicle cost	Total cost	CPU time (sec)	Production cost	Transportation cost	Delay cost	Vehicle cost	Total cost	CPU time (sec)	
1	1854.0	559.7	182.1	500	3095.8	360.5	2014.0	772.7	188.3	700	3675.0	0.4	15.8
2	1786.0	602.1	167.4	500	3055.5	792.4	1946.0	761.7	197.2	650	3555.1	0.1	14.1
3	1845.0	581.0	218.7	500	3144.7	529.5	2005.0	766.6	256.3	700	3727.9	0.2	15.6
4	1849.0	582.8	243.8	500	3175.6	278.3	2069.0	754.0	330.0	650	3803.0	0.3	16.5
5	1750.0	559.2	52.8	450	2812.0	334.2	1860.0	729.3	79.6	600	3268.9	0.3	14.0
6	1862.0	605.0	107.7	550	3124.7	598.2	1968.0	748.9	241.6	700	3658.5	0.3	14.6
7	1808.0	547.4	202.3	500	3057.7	377.0	1985.0	747.4	171.1	750	3653.5	0.1	16.3
8	1841.0	591.9	295.5	500	3228.4	366.3	2081.0	774.7	302.1	700	3857.8	0.1	16.3
9	1859.0	555.1	200.3	500	3114.4	714.6	2009.0	768.0	87.6	700	3564.6	0.1	12.6
10	1782.0	600.5	172.3	500	3054.8	219.7	2032.0	832.6	285.0	800	3949.6	0.2	22.7
Ave.	1823.7	578.5	184.3	500	3086.4	457.1	1996.9	765.6	213.9	695	3671.4	0.2	15.8
(s.d.)	(39.2)	(21.6)	(67.7)	(23.6)	(111.6)	(191.9)	(63.6)	(27.3)	(85.4)	(55.0)	(188.6)	(0.1)	(2.7)

Table 6. Comparison of HGAP and PH&B on type II problems.

Problem	HGAP						PH&B						Total cost reduction rate (%)
	Production Cost	Transportation cost	Delay cost	Vehicle cost	Total cost	CPU time (sec)	Production cost	Transportation cost	Delay cost	Vehicle cost	Total cost	CPU time (sec)	
1	1774.0	529.2	451.2	500	3254.4	1500.0	1824.0	749.2	975.3	600	4148.5	0.4	21.6
2	1738.0	632.1	381.1	550	3301.2	1654.2	1828.0	859.9	1028.1	750	4466.0	0.4	26.1
3	1751.0	520.2	571.5	500	3342.7	1704.8	1861.0	777.3	972.7	650	4261.0	0.5	21.6
4	1738.0	632.1	381.1	550	3301.2	1654.2	1912.0	724.8	926.1	650	4212.9	0.1	21.6
5	1742.0	567.3	414.6	500	3223.9	972.5	1872.0	709.3	935.5	650	4166.8	0.3	22.6
6	1765.0	533.3	658.4	500	3456.7	454.2	1875.0	885.6	1162.0	700	4622.6	0.3	25.2
7	1713.0	540.9	349.5	500	3103.4	2650.0	1813.0	794.6	987.8	600	4195.4	0.3	26.0
8	1730.0	580.9	514.0	500	3324.9	1263.5	1813.0	763.1	1012.2	650	4238.3	0.5	21.6
9	1766.0	534.9	523.3	500	3324.2	1466.3	1866.0	802.1	934.4	700	4302.5	0.4	22.7
10	1765.0	614.5	443.5	550	3373.1	2129.6	1935.0	775.7	1005.4	700	4416.1	0.3	23.6
Ave.	1748.2	568.5	468.8	515	3300.6	1516.7	1859.9	784.2	993.9	665	4303.0	0.3	23.3
(s.d.)	(19.4)	(43.9)	(97.2)	(24.2)	(94.0)	(658.7)	(41.5)	(55.1)	(68.5)	(47.4)	(152.6)	(0.1)	(1.9)

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

of the hard deadline. Other problem input data and genetic parameters are fixed.

Figures 3(a) and 3(b) show variation in the total cost reduction rate and computation time for the number of customers and vehicle capacity, respectively. In Figure 3(a), the reduction rate is increased as N becomes larger. This outcome occurs because HGAP has more opportunity to improve the solutions in its evolution process in the larger problems. In Figure 3(b), the reduction rate is decreased until Q is increased to 600 and, thereafter, the rate is increased. This outcome may occur because HGAP can get more chance to reduce the vehicle cost by combining vehicles when Q becomes smaller below a certain level while generating more routes and can reduce the transportation cost largely by searching better routes when Q becomes larger beyond a certain level while allowing more customer order numbers to be included in a route.

It has been observed in these computational experiments that the solution quality, computation time, and evolution process of HGAP are significantly affected by the geographical distribution of customers, cost conversion factors, and problem type.

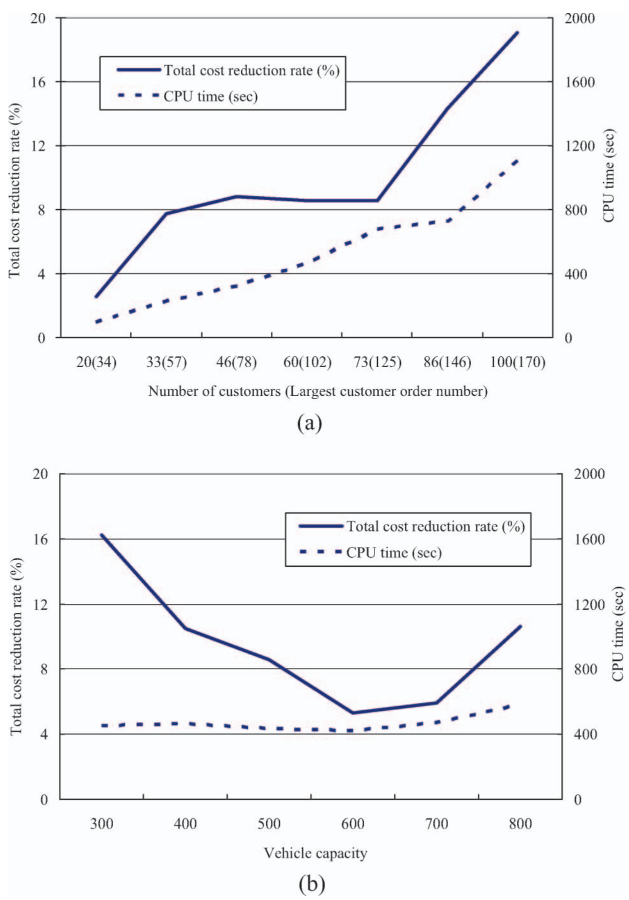

Figure 3. Sensitivity analysis of HGAP for the problem parameters.

6. Conclusion

In this paper, a hybrid genetic algorithm incorporating several local optimisation techniques is proposed for the integrated production and distribution planning of single-period inventory products in a two-echelon supply chain comprised of a single manufacturing facility and many scattered customers with single or multi-item demand. The proposed algorithm, HGAP, was designed to effectively search for a solution that integrates the production sequence and vehicle routes in order to minimise the total cost.

HGAP adopts a mixed scheme of the binary and permutation representation and undergoes continuous evolution while always maintaining solution feasibility. A repair process is performed to correct inefficient routes when necessary. Crossover is conducted separately on three parts of a chromosome, with the BX operator on the first part for the local optimisation algorithm and the PMX operator on either the second part for the production sequence or third part for the vehicle routes. For mutation, random and merge mutations are employed.

The results of the comparison study of HGAP and the mathematical model on small test problems showed that HGAP is capable of reaching solutions that are very close to those obtained by the mathematical model, with an error rate of 0.18% on average. The results of the comparison study of HGAP and the modified Hurter and Buer's heuristic showed that the solution quality provided by HGAP is superior to that of the decoupled planning method, with a total cost reduction rate of 19.6% on average from evolution for 200 generations. In type II problems, a higher cost reduction with HGAP was attained, especially in delay cost, showing that the integrated planning approach is more effective in the problems involving customers with multi-item demand dispersed over the whole region than in those involving customers divided into several zones with the same single-item demand. The results of the sensitivity analysis indicated that the value of integrated planning is higher in larger problems and dependent on the vehicle capacity.

In terms of future research, it will be necessary to compare the solutions obtained by HGAP with the lower bounds of optimal solutions to give a measure of how far they are from the optimal solutions. Furthermore, it is necessary to perform a sensitivity analysis of HGAP for various problem input parameters such as costs, vehicle capacity, and time deadlines.

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014

References

Abdelmaguid, T.F. and Dessouky, M.M., 2006. A genetic algorithm approach to the integrated inventory-distribution problem. *International Journal of Production Research*, 44 (21), 4445–4464.

Alvarenga, G.B., Mateus, G.R., and Tomi, G., 2007. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers and Operations Research*, 34, 1561–1584.

Aytug, H., Khouja, M., and Vergara, F.E., 2003. Use of genetic algorithms to solve production and operations management problems: a review. *International Journal of Production Research*, 41 (17), 3955–4009.

Bräysy, O., Dullaert, W., and Gendreau, M., 2004. Evolutionary algorithms for the vehicle routing problem with time windows. *Journal of Heuristics*, 10, 587–611.

Bräysy, O. and Gendreau, M., 2001. *Genetic algorithms for the vehicle routing problem with time windows*. SINTEF Report, SINTEF Applied Mathematics, Research Council of Norway.

Bräysy, O. and Gendreau, M., 2005. Vehicle routing problem with time windows: Part II. *Metaheuristics*, 39 (1), 119–139.

Buer, M.G.V., Woodruff, D.L., and Olson, R.T., 1999. Solving the medium newspaper production/distribution problem. *European Journal of Operational Research*, 115, 237–253.

Chaudhry, S.S. and Luo, W., 2005. Application genetic algorithms in production and operations management: a review. *International Journal of Production Research*, 43 (19), 4083–4101.

Clarke, G. and Wright, J., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12, 568–581.

Cordeau, J.F., Gendreau, M., Laporte, G., Potvin, J.Y., and Semet, F., 2002. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53 (5), 512–522.

Cunha, C.B. and Mutarelli, F., 2007. A spreadsheet-based optimization model for the integrated problem of producing and distributing a major weekly newsmagazine. *European Journal of Operational Research*, 176 (2), 925–940.

Elhedhli, S. and Goffin, J., 2005. Efficient production-distribution system design. *Management Science*, 51 (7), 1151–1164.

Erenguc, S.S., Simpson, N.C., and Vakharia, A.J., 2001. Integrated production/distribution planning in supply chains: an invited review. *European Journal of Operational Research*, 115, 219–236.

Gen, M. and Syarif, A., 2005. Hybrid genetic algorithm for multi-time period production/distribution planning. *Computers & Industrial Engineering*, 48, 799–809.

Goldberg, D.E. and Lingle, R., 1985. Alleles, loci, and the TSP. *In*: *Proceedings of the First International Conference on Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum Associates, 154–159.

Gottlieb, J., Puchta, M., and Solnon, C., 2003. A Study of greedy, local search, and ant colony optimization approaches for car sequencing problems. *Lecture Notes in Computer Science*, 2611, 245–282.

Hurter, A.P. and Buer, M.G.V., 1996. The newspaper production/distribution problem. *Journal of Business Logistics*, 17 (1), 85–107.

ILOG CPLEX 9.1, 2005. *User's manual*. ILOG S. A., France.

Lei, L., Liu, S., Ruszczynski, A., and Park, S., 2006. On the integrated production, inventory, and distribution routing problem. *IIE Transactions*, 38 (11), 955–970.

Nonino, F. and Panizzolo, R., 2007. Integrated production/distribution planning in the supply chain: the Febal case study. *International Journal of Supply Chain Management*, 12 (2), 150–163.

Park, Y.B., 2005. An integrated approach for production and distribution planning in supply chain management. *International Journal of Production Research*, 43 (6), 1205–1224.

Schmitt, L.J. and Amini, M.M., 1998. Performance characteristics of alternative genetic algorithmic approaches to TSP using path representation: an empirical study. *European Journal of Operational Research*, 108, 551–570.

Shapiro, J.F., 1999. Chap. 23: Bottom-up vs. top-down approaches to supply chain modeling. *In*: S. Tayur, R. Geneshan, and M. Magazine, eds. *Quantitative Models for Supply Chain Management*. Boston: Kluwer Academic Publishers.

Song, S.H., Lee, K.S., and Kim, G.S., 2002. A practical approach to solving a newspaper logistics problem using a digital map. *Computers and Industrial Engineering*, 43, 315–330.

Appendix: Pseudo code of HGAP

```
; Creation of 90% of initial chromosomes
For k = 1 to pop_size*0.9
        string(1, k, 1, 1 to num_local_opt) = 0 or 1
        string(1, k, 2, 1 to num_item) = random(1, num_item)
        string(1, k, 3, 1 to num_order) = random(1, num_order)
Next k
; Creation of 10% of initial chromosomes
For l = 1 to pop_size*0.1
        string(1, k + l, 1, 1 to num_local_opt) = 0 or 1
        string(1, k + l, 2, 1 to num_item) = roulette_wheel
          (K1* K2∧ asc_order_rank(setup_time(item, current_
          prod_seq)) − 1))
        string(1, k + l, 3, 1 to num_order) = roulette_wheel
          (K1* K2∧ dec_order_rank(savings(route_pair)) − 1))
Next l
For g = 1 to num_generations
; Crossover operation
   For n = 1 to pop_size*prob_c
          relative_fitness(k) = (max_total_cost(g)-total
              cost(k))/(max_total_cost(g)-min_total_cost(g))
          parent(g, 1 to 2) = roulette_wheel(relative_fitness(k))
             offspring(num_offspring, 1, 1 to num_local_opt) =
                BX(parent(g, 1), parent(g, 2))
             offspring(num_offspring, 2 or 3, 1 to num_local_
               opt) = PMX(parent(g, 1), parent(g, 2))
             num_offspring = num_offspring + 1
     Next n
; Adaptive random mutation operation
   For n = 1 to pop_size*prob_r*(min_total_cost(g)/
      ave_total_cost(g))
          offspring(num_offspring, 1, 1 to num_
             local_opt) = 0 or 1
          offspring(num_offspring, 2, 1 to num_item) =
             random(1, num_item)
          offspring(num_offspring, 3, 1 to num_order) =
             random(1, num_order)
          num_offspring = num_offspring + 1
     Next n
```

```
; Merge mutation operation
    For n = 1 to num_offsring
        If random_number < prob_g then
            offspring(n, 3, 1 to num_order) = search_and_
              combine(offspring(n, 3, 1 to num_order))
        End if
    Next n
; Construction of the new population after the local
optimisation to the offspring
    For p = 1 to num_offspring
            string(g + 1, p, 1, 1 to num_local_opt) =
              offspring(p, 1, 1 to num_local_opt)
            string(g + 1, p, 2, 1 to num_item) = local_
              opt(offspring(p, 2, 1 to num_item))
            string(g + 1, p, 3, 1 to num_order) = local_
              opt(offspring (p, 3, 1 to num_order))
    Next p
    For p = num_offspring + 1 to pop_size
        string(g + 1, p, 1, 1 to num_local_opt) = string
            (g, best_string(rank), 1, 1 to num_local_opt)
        string(g + 1, p, 2, 1 to num_item) = string
            (g, best_string(rank), 2, 1 to num_item))
        string(g + 1, p, 3, 1 to num_order) = string
            (g, best_string(rank), 3, 1 to num_order))
        rank = rank + 1
    Next p
Next g
```

Downloaded by [University of Tennessee, Knoxville] at 14:13 27 December 2014