International Journal of Applied Operational Research
Vol. 5, No. 2, pp. 35-48, Spring 2015

Journal homepage: ijorlu.liau.ac.ir

An Extended Compact Genetic Algorithm for Milk Run Problem with Time Windows and Inventory Uncertainty

H. Nozari*, A. Aliahmadi , M. Jafari-eskandari, Gh. Khaleghi

Received: 3 October 2014 **;** **Accepted:** 7 March 2015

Abstract In this paper, we introduce a model to optimization of milk run system that is one of VRP problem with time window and uncertainty in inventory. This approach led to the routes with minimum cost of transportation while satisfying all inventory in a given bounded set of uncertainty .The problem is formulated as a robust optimization problem. Since the resulted problem illustrates that grows up time in this method is progressive, and in order to solve the large-scale problems, eCGA (Extended Compact Genetic Algorithm) has been proposed. The efficiency and effectiveness of the eCGA to this optimization problem is tested and the results are presented. The results show the ability of the eCGA to efficiently optimal solution for the cases considered. Investigated example showed that this method has the ability to obtain a solution more accurate than other optimization methods.

1 Introduction

Many industrial activities deal with the problem of routing .The idea of lean production has increasingly become popular and there is a high degree of correlation between a lean production plan and a good logistic strategy. Services have experienced tremendous growth in recent years. For example, SAIPA Company in Iran has sold over $12 billion, and the cost of transport in company in years is over $12 million [1]. Thus, congestion and variation in demand and travel times affects these industries on four main service dimensions:

1) travel time
2) reliability
3) cost of transport
4) cost of inventory [2].

Therefore, there is a need to develop tools for routing and scheduling to direct estimating of uncertainty. The implementation of a good logistic strategy on production planning plays an important role on the success of the whole management. The first step towards bringing the logistics and pure objectives in one line is to handle the supply network control from the suppliers and make it observable. The vehicle routing problem (VRP) is commonly defined as

* **Corresponding Author.** (✉)
E-mail: Hnozari@ind.iust.ac.ir (H. Nozari)

H. Nozari*, A. Aliahmadi ,
Department of Industrial Engineering, Iran University of Science and Technology, Iran

M. Jafari-eskandari
Department of Industrial Engineering, PNU university, Iran

Gh. Khaleghi
Department of Industrial Engineering, Iran University of Science and Technology, Iran

[Downloaded from ijaor.ir on 2026-11-28]

the problem of determining optimal delivery or collection routes from several depots to a set of geographically scattered customers, under a variety of side conditions [3]. The VRP is normally formulated as mixed integer problem (MIP) and there have been many branch and bound method to solve such problem [4]. However, many VRP which are formulated as MIP are classified as NP-hard problem. There are other mathematical methods which are implemented such as set covering, shortest path, etc.[5]. There is a special case of VRP problem called milk run which would be the main focus of this paper [6]. In fact, the modeling determines the exact time that a particular supplier must be ready to ship the order. The other important factor in our model is the ability to diversify orders among suppliers. The proposed model is also capable of assigning different routes for a truck. We present a practical MIP to find the optimal transportation of different parts for an auto industry based on the idea of milk run concept. The resulted problem formulation is run with an actual data using a real world case study. The results are discussed and compared with the actual data. Since the proposed method of the paper is MIP and is hard to solve for a real world problem we also use a metaheuristic approach to solve the resulted problem. The proposed method of this paper is purposely designed for one of the biggest auto maker in the world called SAIPA. According to the company's financial statement for the fiscal year ends in March 2008, the company assembles about half a million cars which is almost half of Iran's market share. The proposed method of this paper is more customized to take into account the management strategy which did not explicitly exist in previous research. In this paper, we propose the milk run problem with uncertain inventory on a set of fixed inventory points with its robust linear counterpart that is robust to parameter uncertainty. In this framework, the perturbations in the return parameters are modeled as unknown, but bounded, and optimization problem is solved assuming worst case behavior of these perturbations. This robust optimization framework was introduced in Alvarenga et al.[7] for linear programming and in Lacomme et al. [8] for general convex programming. This paper uses Extended Compact Genetic Algorithm (eCGA) to solve this optimization problem. eCGA is a different approach from the conventional GA and uses probabilistic model building concept to solve the linkage learning problem [9]. Linkage Learning in GA is to find the building blocks which should be preserved after crossover and then make GA work with found building Blocks. The probability model used in eCGA, namely Marginal Product Model (MPM) is distinct from other previously described models in the sense that MPM includes both univariate as well as multivariate marginal distribution in its probability model. It assumes no directional dependency among variables and takes in account marginal probability of set of variables at once. eCGA uses greedy search to find the good MPM model and runs compact GA on that model.

In what follows ,the literature is first reviewed, then milk run problem with time windows with inventory uncertainty is formulated. Basics of the eCGA is then presented in the next section. Finally, the results are shown and compared with actual solutions.

2 Review of Literature

Problems where a given set of vehicles with finite capacity have to be routed to satisfy a geographically dispersed demand at minimum cost are known as VRPs. VRP belongs to the typical complicated combination optimization problem, as a NP difficult problem. VRP has close relation with travelling-salesman problem (TSP). Some scholars consider VRP as the combination of bin packing problem (BPP) and TSP [10]. There are many kinds of VRP models which consider classification in structure bottom (Fig 1).

[Downloaded from ijaor.ir on 2026-11-28]

Milk run system determines the route, the time schedule, the type and the number of parts that different trucks must choose in order to receive the orders from various suppliers with the primary assumption that all trucks must return the empty pallets to the demand centre (e.g., auto maker). There are different objectives involved in this kind of modeling which need to be minimized such as inventory and transportation costs.

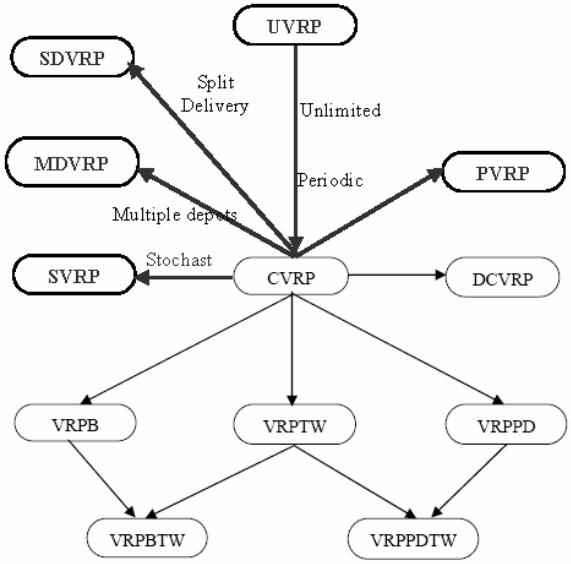

Fig. 1 All VRP

The cost advantages of this system for short distances and consignments with high delivery frequency and value is extremely remarkable and we will show this with some actual data. As the distance between the suppliers and the manufacturers increases, the advantages of performing this system increase at a declining speed. Also through increasing such parameters like suppliers geographical density, value of parts, expenses for keeping the inventory and production circulation, the advantages of running this system increase. In this system, the frequency of the consignments delivery depends on the price and the size of the parts. Large size and valuable parts have to be transported with more frequency to prevent increasing of the inventory expenses. In this system, the company, based on MRP system output, determines the weekly requirements of every supplier and the time schedule. We also determine the coherence and route of collecting the parts. Running pure logistic network requires three principal changes in the system:

1. reducing the number of orders or stockpiles
2. increasing the number and the frequency of delivery to the factory
3. making a smooth course of materials inflow to the factory the most important

Advantages of running milk run system include:

- materials and parts inflow to the production line is becoming easier
- the performance of the supply chain and the logistic is improved due to effectively using of the transporting vehicles' spaces, controlling the transport charges as well as reducing the level of parts inventory and their maintenance costs
- the space of the valuable warehouse in a factory is reduced

[Downloaded from ijaor.ir on 2026-11-28]

- the total number of required pallets in the supply chain is reduced and the capital expenditures, maintenance and repairs costs and operating expenses of the pallets are less than the other methods
- it reduces the lack of confidence at the time of delivery of parts and consequently reduces the parts stock level
- it increases the capital turnover
- it increases the flexibility in supplying the parts
- it soothes and establishes a good discipline in logistic operations (e.g., loading and unloading).

There are numerous factors which may influence the features of the logistic patterns and make a relative superiority of some of these patterns over other ones [11]. For example, the lack of readiness of information infrastructures may lead to priority of direct shipment pattern over other ones, because the reliance of this pattern on the information infrastructures is not so much vital. Also, the low level of the production or when the supplied parts have a small size may be strength for those patterns such as shipment through storerooms or collecting from suppliers and shipment to the manufacturing company, because in this situation, other patterns impose high level of inventory on the system. Addressing data uncertainty in mathematical programming models has long been recognized as a central problem in optimization. There are two principal methods that have been proposed to address data uncertainty over the years:

a. stochastic programming
b. robust optimization.

As early as the mid-1950s, Dantzig and Ramser [12] introduced stochastic programming as an approach to model data uncertainty by assuming scenarios for the data occurring with different probabilities. The two main difficulties with such an approach are:

a knowing the exact distribution for the data b the size of the resulting optimization model increases drastically as a function of the number of scenarios, which poses substantial computational challenges [13]. Robust optimization refers to the modeling of optimization problems with data uncertainty to obtain a solution that is guaranteed to be good for all or most possible realizations of the uncertain parameters. In recent years, robust optimization has gained substantial popularity as a competing methodology for solving several types of stochastic optimization models. Robust optimization has been successful in immunizing uncertain mathematical optimization. The first step in this direction is taken by Soyster [14]who proposes a worst case model to linear optimization. Subsequently, more elaborate uncertainty sets and computationally attractive robust optimization methodologies are proposed by El-Ghaoui and Lebret [15], and El-Ghaoui et al. [16], Goldfarb and Iyangar [17], Bertsimas and Sim [18][19]. To address the issue of over-conservatism in robust linear optimization, these papers propose less conservative models by considering uncertainty sets in the form of ellipsoidal and more complex intersection of ellipsoidal sets. The robust counterparts of the nominal problems generally are in the form of conic quadratic. The goal of this paper is to present such an approach, based on robust optimization which has recently become an interesting research topic. These ideas are ported to a mathematical programming context beginning with the work by Ben-Tal and Nemirovski[20][21] where the authors formulate the robust optimization problems of linear programs, quadratic programs, and general convex programs. We utilize the approach which leads to robust counterparts while controlling the level of conservativeness of the solution. Independently, El-Ghaoui et al.[16]

[Downloaded from ijaor.ir on 2026-11-28]

study the same robust optimization counterpart for semi-definite programming problems. More recently, this approach is extended to integer programming problems.[26]

3 Robust Milk Run And Time Windows

In this section, we first identify formulation of the deterministic milk run with time windows and inventory uncertainty sets that will be used and then, we will offer a derive for the robust milk run with time windows.

3.1 proposed model for milk run with time windows

In this section, we deal with the milk system with time windows modeling. As we already explained, the proposed method of this paper is a special case of milk run which is purposely designed and customized for SAIPA auto maker. In fact, there are three main issues which need to be handled in the proposed method. The first requirement is to use different suppliers to provide some particular parts. In fact, the company monitors the service level for some selected suppliers and based on the quality of their services, management makes some necessary decisions to adjust the quantity of each vendor. The other problem is that the model must determine the exact time that a supplier would ship the parts. In other word, every supplier needs to know the exact time to ship a particular part and this could help the system to pre-schedule the production plan. Finally, the proposed model of this paper is capable of changing a truck to work in different routes in various time schedules. In the following section, we first introduce the decision variables and the parameters used in our problem formulation.

3.1.1 Decision variables and parameters

$x_{t_j kpj}$ the number of shipping palates of part p transported with vehicle k from supplier j with the due date of time t_j

V_k truck capacity k

$$y_{tkij}: \begin{cases} 0 & if \ \ x_{t_j kpj} > 0 \\ 1 & if \ x_{t_j kpj} = 0 \end{cases}$$

$c_p^{\min}$ the minimum inventory level of part p

V_p^{PL} the maximum capacity of palate for part p

U_{tp} the average consumption of part p in time t

X_{tp}^{M} the remaining part p at the end of time t

H_p the inventory cost per hour of each palate containing part p in warehouse

C_{ikj} the cost of the truck k moving from supplier i to supplier j

γ_{pj} the percentage of part p allocated to supplier j

[Downloaded from ijaor.ir on 2026-11-28]

$\overset{\circ}{C}$ the fixed cost of waiting a truck at each supplier
S_i the service time required at supplier *i*
T_{ij} the time needed to travel from supplier *i* to supplier *j*
B_k the departure time of vehicle *k* at the distribution center
r_k the required returning time of vehicle *k* to the distribution center.

3.1.2 Objective function

Since the main goal of implementation of the milk run system with time windows is to decrease the transportation costs and reduce the level of inventory of parts at the warehouse we consider the following objective function:

$$\min \sum_t \sum_k \sum_i \sum_j [C_{kij} + C'] Y_{tkij} + \sum_t \sum_p H_p \times X_{tp}^M + \sum_i P_i(t_i)$$

The first part of the objective function explains the total costs of transportation. This function states that if a transporting vehicle moves from one supplier to another one, the transportation cost with the vehicle and the fixed cost of the loading must be considered. The second part is used for inventory costs. In summary, the basis of the objective function is to simultaneously reduce the transportation costs as well as the inventory expenditures. The tertiary part is the penalty costs. The penalty function is formally defined by equation (1).

$$P_i(t_i) = \begin{cases} \phi & if \quad t_i \prec e_i \\ f_i^e + U_e(a_i - t_i) & if \quad e_i \le t_i \prec a_i \\ 0 & if \quad a_i \le t_i \le b_i \\ f_i^l + U_l(t_i - b_i) & if \quad b_i \prec t_i \le l_i \\ \phi & if \quad t_i \ge l_i \end{cases} \tag{1}$$

where t_i is the time a vehicle arrives at supplier *i*, and f_i^e, f_i^l are the respect minimum(fixed) costs when waiting or lateness occurs, U_e is the waiting cost per unit of time, and U_l is the lateness penalty per unit of time.
The first constraint is associated with truck capacity.

$$\sum_p \sum_j X_{t_j kpj} \times V_{pl}^p \le V_k \qquad \forall (t_j, k) \tag{2}$$

This constraint investigates that the number of palates collected from the suppliers for transportation to warehouses from a proper volume of parts for collecting and transporting to warehouses would not exceed the number of transporting vehicles.

$$\sum_{t_j} \sum_k X_{t_j kpj} = \gamma_{pj} \qquad \forall (p, j) \tag{3}$$

[Downloaded from ijaor.ir on 2026-11-28]

This constraint states that every supplier is only allowed to transfer the volume of parts committed in the contract. This constraint is stated here because in the companies who are relying on the suppliers, usually supply their similar type parts from two or several suppliers.

$$\sum_{t_j}\sum_{k} X_{t_j kpj} = \gamma_{pj} \qquad \forall(p,j) \tag{4}$$

Each supplier may set a minimum inventory level for its own parts which is define according to various factors. This amount of the inventory is usually determined for some working days (two or three days).

$$X^{M}_{tp} = \sum_{k}\sum_{j} X_{t_j kpj} + X^{M}_{(t-1)p} - U_{tp} \qquad \forall(t,p) \tag{5}$$

The total number of each parts is equal to the total number of parts arrive at the end of the time t, plus the inventory of the parts at the end of time $t-1$ minus the number of parts used in time t. This constraint studies that in each moment what quantity of the parts are available to avoid shortage of the parts in the warehouse.

$$\sum_{k}\sum_{j} Y_{tkij} \le 1 \qquad \forall j \ge 2, \forall t \tag{6}$$

$$\sum_{k}\sum_{i} Y_{tkij} \le 1 \qquad \forall i \ge 2, \forall t \tag{7}$$

These constraints use one and only one transporting vehicle for each time schedule t. In this model it is presumed that two or several transporting vehicles may not be used,

Simultaneously. In this model, the warehouse of the manufacturing company has been marked with $i = 1$ number. Note that since the company's warehouse is also considered as a supplier we must separate these two constraints. Therefore, this issue is explained in the next two constraints. The first constraint states that in one moment, at most one transporting vehicle can enter to supplier j from the supplier i and in the next constraint, only one transporting vehicle can exit the supplier i.

$$\sum_{j} Y_{tk1j} \le 1 \qquad \forall(t,k) \tag{8}$$

$$\sum_{i} Y_{tki1} \le 1 \qquad \forall(t,k) \tag{9}$$

Constraints (8) and (9) are similar to (6) and (7) but the difference is that these constraints are exclusively related to the warehouse of the manufacturing company.

$$\sum_{i} Y_{tkiq} = \sum_{i} Y_{tkqj} \qquad \forall(t,k), \forall q > 1 \tag{10}$$

This constraint which is designed for sequencing the routes states that if a vehicle gets to a knot, it must exit from it. The purpose of this constraint is to avoid the stoppage of the vehicle in the place of one supplier and each transporting vehicle, entering any place, must exit from

[Downloaded from ijaor.ir on 2026-11-28]

that place. Note that the transporting vehicle only stops when it arrives at the supplier's warehouse, and this condition will be stated in the following constraint:

$$\sum_{i\in S}\sum_{j\in S} Y_{tkij} \le |S| - 1 S \subseteq \{2,3,...,KT\} \quad \forall (t,k) \tag{11}$$

(*KT* is the total number of the suppliers.)

This relation states that the each transporting vehicle starts from the supplier's warehouse and its destination is the same warehouse. This constraint is stated hereto avoid creation of laps in the transportation route. This constraint states that the transporting vehicle never returns to the place it has referred to before in this route.

$$X_{t_j kpj} \le M \times \sum_i Y_{t_j kij} \qquad \forall (t_j, k, p, j) \tag{12}$$

(*M* is a big number.)

The constraint (12) states that we can bring parts from the supplier *j* if the transporting vehicle is able to enter from one supplier to this supplier. This constraint is important because it states the relationship between two alternatives of *x* and *y*.

$$t_j \ge Max\{t_i, a_i\} + S_i + T_{ij} - M(1 - X_{t_j kpj}) \tag{13}$$

(*M* is a big number.)

The precedence relation between two successive nodes is expressed.

$$B_k + \sum_i \sum_j \{X_{t_j kpj} Max\{t_j, a_j\} + S_j\} \le r_k \quad \forall k \tag{14}$$

The required returning time of each vehicle is constrained and equation (1).

3.2 Uncertainty in inventory

We consider that the minimum inventory level of part *p*, parameter $c_p^{\min}$ is uncertain and belongs to a bounded set *U*. We consider uncertainty sets which are constructed as deviations around an expected minimum inventory value $c_p^{\min^0}$.The possible deviation directions from these nominal values are fixed and identified by scenario vectors, $c_p^{\min^k} \in \Re^N$,where *N* is the number of nodes. The scenario vectors are allowed to have negative deviation values (Sungur et al., 2008). For a given number of scenario vectors, *n*, the general uncertainty set *U* is a linear combination of the scenario vectors with weights $g \in \Re^n$ that must belong to a bounded set $g \in G$:

$$U_C = \left\{ c_p^{\min} \middle| c_p^{\min^0} + \sum_{k=1}^{n} g_k c_p^{\min^k},\ g \in G \right\}$$

In particular, we consider the following set for *G* (Ben-Tal and Nemirovski, 1999):

[Downloaded from ijaor.ir on 2026-11-28]

$$\textit{convex hull} \quad G = \left\{ g \in \Re^N \middle| g \geq 0, \sum_{k=1}^{n} g_k \leq 1 \right\}$$

3.3 Robust milk run with time windows formulation

We now propose the robust counterpart problem milk run with minimum inventory belonging to an uncertainty set U. Recall that we consider the problem only with uncertainty in constraint (4). We can therefore state the robust milk run. This problem minimizes objective, subject to constraints (1), (2), (3), (5), (6), (7), (8), (9), (10), (11), (12), (13) and (14). If we substitute in the definition of the uncertainty set U_C, we can write the robust constraint (4') as the following inequality:

$$X_{tp}^{M} - c_p^{\min^0} \geq \sum_{k=1}^{n} g_k c_p^{\min^k}, \quad \forall g \in G, \forall (t,p) \qquad (4')$$

4 Experimental Analysis

In order to analyze the performance of the proposed method, we have first solved the resulted model using mixed integer software package using some real data from SAIPA complex for the group of variables: the number of suppliers, the number of parts and the time horizon for planning. Since there are many binary variables in our proposed model we may not be able to run the resulted model for large problems. Therefore, we use a robust optimization to solve the problem. Table 1 shows the details of the implementation of both methods. As we can observe, there is no difference between the optimal solutions of the proposed method for some small-scale problems. However, as the size of the applications grows the gap between the optimal solution and the robust optimization is getting bigger. On the other hand, the CPU time needed to solve the proposed method is relatively acceptable for some small problems but for real world case problems we may not be able to find the optimal solution very easily. All experiments were carried out with a runtime limit of an hour on a COMPAQ EVo N1020v computer with a 2.4 GHz Intel Xeon Processor and 2 GB RAM running Red Hat Windows XP.

As Table 1 shows, sensitivity increases when the number of suppliers and uncertainty go up. CPU time, in this method, grows progressively and it is necessary to use a heuristic or Meta heuristic method to solve the large-scale problems. These results show that although robust solution increase some transportation and inventory cost, there are many advantages to meet the customers, such as decreasing the inventory in the warehouse as direct outcome and loyalty growth as an indirect parameter. Since the resulted problem illustrate that grows up time in this method is progressively, In order to solve the problem in large-scale, eCGA has been proposed.

Table 1 The computational results of deterministic and robust optimization

The number of		Time	The proposed deterministic		The robust optimization	
Supplier	Part	horizon	CPU	Cost	CPU	Cost
2	5	5	0.3	3.6E + 4	0.45	4.1E + 4
2	5	10	0.37	7.2E + 4	1.06	9.32E + 4
3	10	15	2.37	32E + 4	4.43	39.3E + 4
3	10	20	5.47	41 + E4	8.45	50.1E + 4
4	15	30	10.58	129 + E4	15.1	141E + 4

[Downloaded from ijaor.ir on 2026-11-28]

The number of		Time	The proposed deterministic		The robust optimization	
Supplier	Part	horizon	CPU	Cost	CPU	Cost
4	15	40	17.85	172E + 4	23.25	197E + 4
5	20	60	37.5	384E + 4	45.12	432E + 4
5	20	80	68.95	492E + 4	84.32	545E + 4
6	25	150	157.98	106E + 5	205.55	127E + 5
6	25	200	453.87	135E + 5	764	205E + 5
7	30	280	1026.35	253E + 5	1798.35	352E + 5
7	30	360	2845.63	316E + 5	4050.45	424E + 5

5 Extended Compact Genetic Algorithm

The extended compact GA proposed by Harik [22] is based on a key idea that the choice of a good probability distribution is equivalent to linkage learning. The measure of a good distribution is quantified based on minimum description length (MDL) models. The key concept behind MDL models is that all things being equal, simpler distributions are better than more complex ones. The MDL restriction penalizes both inaccurate and complex models, thereby leading to an optimal probability distribution. Thus, MDL restriction reformulates the problem of finding a good distribution as an optimization problem that minimizes both the probability model as well as population representation. The probability distribution used in eCGA is a class of probability models known as marginal product models (MPMs) shown schematically in Fig. 2. MPMs are formed as a product of marginal distributions on a partition of the genes and are similar to those of the compact GA (CGA) [23] and PBIL [24]. Unlike the models used in CGA and PBIL,MPMs can represent probability distributions for more than one gene at a time. MPMs also facilitate a direct linkage map with each partition separating tightly linked genes. For example, the following MPM, (1,3)(2)(4), for a four-bit problem represents that the 1st and 3rd genes are linked and 2nd and 4th genes are independent. Additionally, the MPM consists of the marginal probabilities:

$$\{p(x_1=0,x_3=0),p(x_1=0,x_3=1),p(x_1=1,x_3=0),p(x_1=1,x_3=1),p(x_2=0),p(x_2=1),$$
$$p(x_4=0),p(x_4=1)\}$$

where x_i is the value of the its gene. The eCGA can be algorithmically outlined as follows:

1. Initialization: The population is usually initialized with random individuals. However, other initialization procedures can also be used.
2. Evaluate the fitness value of the individuals
3. Selection: Use a selection mechanism to select a sub-population out of the current population. The ECGA uses ranking selection. However, other selection procedures can be used instead of ranking selection. The sub-population will be used to create the MPM defined later. The size of the sub-population is referred to as tournament size in this work.

[Downloaded from ijaor.ir on 2026-11-28]

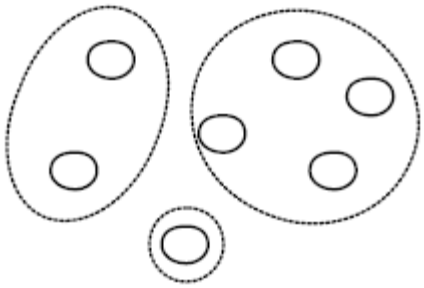

Fig. 2 marginal product model (MPM)

4. Build the probabilistic model: In eCGA, both the structure and the parameters of the model are searched. A greedy search heuristic is used to find an optimal model of the selected individuals in the population.
5. Create new individuals: In eCGA, new individuals are created by sampling the probabilistic model.
6. Replace the parents with the offspring.
7. Repeat steps 2{6 until some termination criteria are met.

Two things need further explanation. one is the identification of MPM using MDL and the other is the creation of a new population based on MPM. The identification of MPM is formulated as a constrained optimization problem,

$$\text{Min} \quad C_m + C_p \tag{15}$$

s.t.

$$2^{k_i'} \le n \qquad \forall i \in [1, m] \tag{16}$$

where C_m is the model complexity which represents the cost of a complex model. In essence, the model complexity, C_m, quantifies the model representation size in terms of number of bits required to store all the marginal probabilities. Let, a given problem of size l with binary alphabets, have m partitions with k_i genes in the ith partition, such that $\sum_{i=1}^{m} k_i' = l$. Then each partition i requires $2^{k_i'} - 1$ independent frequencies to completely define its marginal distribution. Furthermore, each frequency is of size $\log_2(n)$, where n is the population size. Therefore, the model complexity C_m, is given by

$$C_m = \log_2(n) \sum_{i=1}^{m} (2^{k_i'} - 1) \tag{17}$$

The compressed population complexity, C_p, represents the cost of using a simple model as against a complex one. In essence, the compressed population complexity, C_p, quantifies the data compression in terms of the entropy of the marginal distribution over all partitions. Therefore, C_p is evaluated as

$$C_p = \log_2(n) \sum_{i=1}^{m} \sum_{j=1}^{2^{k_i'}} - p_{ij} \log_2(p_{ij}) \tag{18}$$

[Downloaded from ijaor.ir on 2026-11-28]

Where p_{ij} is the frequency of the jth gene sequence of the genes belonging to the *i*th partition. In other words, $p_{ij} = N_{ij} / n$, where N_{ij} is the number of chromosomes in the population (after selection) possessing bit-sequence $j \in \left[1, 2^{k_i'}\right]$ for *i*th partition. The constraint (Equation 16) arises due to finite population size.

The following greedy search heuristic is used to find an optimal or near-optimal probabilistic model:

1. Assume each variable is independent of each other. The model is a vector of probabilities.
2. Compute the model complexity and population complexity values of the current model.
3. Consider all possible $\frac{1}{2}l(l-1)$ merges of two variables.
4. Evaluate the model and compressed population complexity values for each model structure.
5. Select the merged model with lowest combined complexity.
6. If the combined complexity of the best merged model is better than the combined complexity of the model evaluated in step 2, replace it with the best merged model and go to step 2.
7. If the combined complexity of the best merged model is less than or equal to the combined complexity, the model cannot be improved and the model of step 2. is the probabilistic model of the current generation.

The offspring populations are generated by randomly choosing subsets from the current individuals according to the probabilities of the subsets as calculated in the probabilistic model [25].

In this problem the output of ECGA algorithm, which have been obtained using some real data , are shown in table 2. According to the outputs, for problem, there is a slight difference between the answers to this algorithm and the answers to other methods. The gap between the answer of this algorithm and the actual plan change from 0.5% to 5% .

Table 2 The computational results of MIP for robust optimization and eCGA algorithm for robust optimization

The number of		Time horizon	The proposed MIP method for robust optimization		The proposed eCGA algorithm for robust optimization	
Supplier	Part		CPU	Cost	CPU	Cost
2	5	5	0.45	4.1E + 4	0	4.1E+4
2	5	10	1.06	9.32E + 4	0	9.32E+4
3	10	15	4.43	39.3E + 4	0	39.3E+4
3	10	20	8.45	50.1E + 4	0.01	50.1E+4
4	15	30	15.1	141E + 4	0.04	141.1E+4
4	15	40	23.25	197E + 4	2.5	197E+4
5	20	60	45.12	432E + 4	15.1	432.1E+4
5	20	80	84.32	545E + 4	41.5	545.2E+4
6	25	150	205.55	127E + 5	115.1	127.1E+5
6	25	200	764	205E + 5	235.55	207.5E+5
7	30	280	1798.35	352E + 5	572.25	362E+5
7	30	350	4050.45	424E + 5	1293.55	424E+5

6 Conclusions

In this study, we have proposed a deterministic, numerically tractable methodology to

[Downloaded from ijaor.ir on 2026-11-28]

address a new problem of optimal controlling supply chains, milk run system with time windows, subject to uncertain inventory. We propose the use of robust optimization to obtain efficient routing solutions for problems under uncertainty. Our work has shown that robust optimization is an attractive alternative for formulating milk run problem under uncertainty since it does not require distribution assumptions on the uncertainty or a cumbersome representation through scenarios.

This method uses very little information on the uncertain inventory. We present computational results, which investigate some sample test problems. Our computational results show that the robust solution can be protected from unmet inventory while incurring a small additional cost over deterministic optimal routes. Results of our investigation show that although the robust solution imposes extra cost to the transportation and inventory warehouse, it does not remain any unmet inventory in the network so it was more efficient than the deterministic solution. The method which we used is an exact and is not applicable for large-scale problems but it is a proper tool to validate the heuristic algorithms. Finally, we have discussed about some affective parameters and their role in the robust solution.

References

1. Jafari, M., Hassan Zade, F., (2009). Costs of inventory & maintenance in SAIPA Company. Technical report, IRAN County.
2. Meyer Mohaddes Associates, Inc. 2006. Gateway cities trucking study. Technical report, Gateway Cities Council of Governments Southeast Los Angeles County.
3. Kallehauge, B., 2008. Formulations and exact algorithms for the vehicle routing problem with time windows. Computers and Operations Research, Vol. 35, pp.2307–2330.
4. Toth, P., Vigo, D., (2002). Branch-and-bound algorithms for the capacitated VRP. The Vehicle Routing Problem SIAM Monographs on Discrete Mathematics and Applications, pp.29–51, Philadelphia.
5. Christofides, N., Mingozzi, A., Toth, P., (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. Mathematical Programming, Vol. 20, pp.255–282.
6. Jeon, G., Leep, H. R., Shim, J. Y., (2007). A vehicle routing problem solved by using a hybrid genetic algorithm. Computers & Industrial Engineering, Vol. 53, pp.680–692.
7. Mateus, G.R., de Tomi, G., (2007). A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. Computers & Operations Research, Vol. 34, pp.1561–1584.
8. Lacomme, P., Prins, C., Sevaux, M., (2006). A genetic algorithm for a bi-objective capacitated arc routing problem. Computers & Operations Research, Vol. 33, pp.3473–3493.
9. Thierens, D., Goldberg, D. E., (1993). Mixing in genetic algorithms. Proceedings of the Fifth International Conference on Genetic Algorithms (pp. 3845), 1993.
10. Lenstra, J. K., Rinnooy Kan, A. H. G., (1981). Complexity of vehicle routing and scheduling problems. Networks, Vol. 11, pp.221–227.
11. Du, T., Wang, F. K., Lu, P., (2007). A real-time vehicle-dispatching system for consolidating milk runs. Transportation Research Part E, Vol. 43, pp.565–577.
12. Dantzig, G. B., Ramser, J. H., (1959). The truck dispatching problem. Management Science, Vol. 6, pp.80–91.
13. Bertsimas, D., Sim, M., (2004). The price of robustness. Operations Research, Vol. 52, No. 1, pp.35–53.
14. Soyster, A. L., (1957). Convex programming with set-inclusive constraints and applications to inexact linear programming. Operational Research, Vol. 21, pp.1154–1157.
15. El-Ghaoui, L., Lebret, H., (1997). Robust solutions to least-square problems to uncertain data matrices. SIAM J. Matrix Anal. Appl., Vol. 18, pp.1035–1064.
16. El-Ghaoui, L., Oustry, F., Lebret, H., (1998). Robust solutions to uncertain semi definite programs. SIAM J. Optim., Vol. 9, pp.33–52.

[Downloaded from ijaor.ir on 2026-11-28]

17. Goldfarb, D., Iyengar, G., (2003). Robust quadratically constrained programs. Math. Prog. Ser.B, Vol. 97, No. 3, pp.495–515.
18. Bertsimas, D., Sim, M., (2003). Robust discrete optimization and network flows. Mathematical Programming, Vol. 98, Nos. 1–3, pp.49–71.
19. Bertsimas, D., Sim, M., (2004). The price of robustness. Operations Research, Vol. 52, No. 1, pp.35–53.
20.]Ben-Tal, A., Nemirovski, A., (1998). Robust convex optimization. Mathematics of Operations Research, Vol. 23, No. 4, pp.769–805.
21. Ben-Tal, A., Nemirovski, A., (1999). Robust solutions to uncertain programs. Operations Research Letters, Vol. 25, pp.1–13.
22. Harik, G., (1999). Linkage learning via probabilistic modeling in the ECGA. Urbana, IL: University of Illinois Genetic Algorithms Laboratory ,IlliGAL Report No. 99010.
23. Harik, G. R., Lobo, F. G., Goldberg, D. E., (1998). The compact genetic algorithm. In Proceedings of the IEEE Conference on Evolutionary Computation 1998 (ICEG'98) (pp. 523528). Piscataway, NJ: IEEE Service Centre.
24. Baluja, S., (1994). Populationbased incremental learning: A method for integrating genetic search based function optimization and competitive learning. Pittsburgh, PA: Carnegie Mellon University, Technical Report No. CMUCS94163.
25. Sastry, K., Goldberg, D. E., (2004). Designing competent mutation operators via probabilistic model building of neighborhoods. Proceedings of the Genetic and Evolutionary Computation Conference, 2 , 114–125. Also IlliGAL Report No. 2004006.
26. Jafari-Eskandari, M., Aliahmadi, A. R., Khaleghi, G. H. H., (2010). A robust optimisation approach for the milk run problem with time windows with inventory uncertainty: an auto industry supply chain case study. Int. J. of Rapid Manufacturing 2010 - Vol. 1, No.3 pp. 334 - 347

[Downloaded from ijaor.ir on 2026-11-28]