# Multi-objective Inventory Routing with Uncertain Demand Using Population-based Metaheuristics

Zhiwei YANG, Michael EMMERICH, Thomas BÄCK and Joost KOK [*]

*Leiden Institute of Advanced Computer Science, Leiden University, Niels Bohrweg 1, 2333 CA Leiden, the Netherlands*

**Abstract.** This article studies a tri-objective formulation of the inventory routing problem, extending the recently studied bi-objective formulation. As compared to distance cost and inventory cost, which were discussed in previous work, it also considers stockout cost as a third objective. Demand is modeled as a Poisson random variable. State-of-the-art evolutionary multi-objective optimization algorithms and a new method based on swarm intelligence are used to compute approximation of the 3-D Pareto front. A bench-mark previously used in bi-objective inventory routing is extended by incorporating a uncertain demand model with an expected value that equals the average demand of the original benchmark. The results provide insights into the shape of the optimal trade-off surface. Based on this the dependences between different objectives are clarified and discussed. Moreover, the performances of the four different algorithmic methods are compared and due to the consistency in the results, it can be concluded that a near optimal approximation to the Pareto front can be found for problems of practically relevant size.

Keywords: Inventory Routing, Cooperative Swarm Algorithm, Multicriteria Optimization, Robust Optimization, Indicator-based Evolutionary Algorithms

## 1. Introduction

The Inventory routing problem (IRP) is a very important problem in logistics, especially for the vendor managed inventory (VMI) replenishment [36,38]. Many companies are looking for a win-win strategy for the supplier and customers by integrating inventory management, vehicle routing and delivery strategies [13]. This paper considers a finite horizon tri-objective stochastic IRP with a single supplier and multiple geographically distributed customers. In this problem, each customer has a uncertain demand each day for a single product. The customers are replenished from the central supplier by a fleet of homogeneous vehicles with limited capacity. The goal is to find a delivery strategy and routing schedule that can simultaneously minimize three objectives: routing cost, inventory cost and stockout cost.

Many variants of IRP have been developed since the first pioneer paper of inventory routing thirty years ago [19]. The IRP was regarded as an extension of the vehicle routing problem (VRP) in the early papers, and only the routing cost was considered, while the inventory levels were regarded as fixed constraints that had to be high enough to satisfy the stochastic period demand of customers [7]. Burns et al. integrated inventory cost as another objective and analyzed the trade-offs between inventory cost and routing cost with finite time horizon [12,25]. But the time horizon can also be infinite which can be seen in [3,29]. To simplify the problem, direct routing is adopted, which means one vehicle replenishes only one customer in each period [6,29,38]. However, most real world problems are not using direct routing. Recent studies focused on the problem in which one vehicle is able to serve several customers each time [8,21]. During the execution of inventory routing

[*] Corresponding Author: Joost Kok, LIACS, Leiden University, Niels Bohrweg 1, 2333-CA Leiden, the Netherlands; E-mail: j.n.kok@liacs.leidenuniv.nl

a stockout may occur. This can be avoided by adding a back-logging strategy in case customers are willing to wait till the next delivery day [15,17]. Stockout is not permitted in most IRP with deterministic demand [2,13,31]. However, when the demand of customers is stochastic, it is impossible to avoid the stockout [33]. Hence, expected stockout should be regarded as the third objective to be minimized of the inventory routing problem.

The inventory routing problem with two objectives has been approached in different methods ranging from exact algorithms to metaheuristics algorithms. Archetti et al. implemented a branch-and-cut algorithm for a small scale single vehicle IRP with a short time horizon [5]. Solyali et al. improved the branch-and-cut algorithm to an extended small scale single vehicle IRP using a strong formulation [53]. In this extension, each customer has an external dynamic demand and is controlled by a deterministic order-up-to-level policy. Recently, this algorithm has already been implemented to solve larger scale IRP with multiple vehicles [1,20]. However, exact algorithms are only able to solve the small scale IRP within limited time. In order to obtain high quality solutions for large scale IRP problems, metaheuristics algorithms are implemented in the current research. Zhao et al. proposed a variable neighborhood search for the IRP in a three echelon logistics system [57]. A tabu search algorithm combined with ad hoc designed mixed integer programming models was applied by Archetti et al. for an IRP in discrete time [4]. Ribeiro et al. introduced an iterative local search algorithm to solve the IRP with stochastic and deterministic demand [46]. They decomposed the IRP into an each day VRP, and the iterative local search algorithm was used to find a good feasible of the VRP. Salvesbergh et al. combined an improved branch-and-cut algorithm and greedy heuristic which is used to solve the IRP with continuous moves [49]. A hybrid genetic algorithm was developed for a finite horizon, multi-periods, multi-products and many-to-one distribution IRP by Moin et al. [42]. The authors used an allocation first and route second strategy to construct a solution. New crossover and mutation operators and new presentations are introduced in order to adapt the algorithm to the IRP. However, these papers all focus on bi-objective IRP. The thri-objective IRP has not been studied well yet. A modified ant colony algorithm was developed for multi-item inventory routing problem with demand uncertainty [33]. They proposed a new pheromone updating rule which could integrate the stockout cost. But they integrate three objectives problem into a single objective problem by giving a tradeoff weights to each objective. Geiger et al. developed a local search strategy by modifying deliver frequencies which generated a Pareto front approximation for the bi-objective inventory routing problem [31]. They also proposed 14 benchmark instances with certain demand.

Multi-objective optimization problems become a hot topic in the recent years [14,36,40]. The goal in this paper is to reduce three different cost factors simultaneously - routing cost, inventory cost and stockout cost. Whereas the first is related to the economical and ecological aspect of the problem (fuel consumption), the latter is related to the quality of service that needs to be optimized, the last one indicates the lost sale that should be minimized. The goal of this study is to compare two different, though closely related, strategies for computing the Pareto optimal front of this problem.

Many algorithms have been implemented to solve multi-objective optimization problems [16,48,51,58]. In this study, four multi-objective optimization algorithms are compared to solve the tri-objective inventory routing problem with uncertain demand in logistics. These are two state-of-the-art evolutionary multi-objective optimization methods, namely NSGA-II and SMS-EMOA. Moreover, an extension of a state of the art bi-objective inventory routing method by Huber, Geiger and Sevaux [34] for two objectives (inventory and routing cost) is developed to the tri-objective problem. Since the hypervolume indicator is a good measure for the quality of the Pareto front and particle swarms algorithms works well in optimization problems. In this paper we also propose a new hypervolume indicator based particle swarm optimizer called multi-objective optimization cooperative particle swarms (MOCOPS).

The paper is structured as follows: In Section 2 of this paper the inventory routing problem is defined and preliminaries from literature are described. Section 3 demonstrates the structure of the solver and provides a detailed description of the four algorithms. In section 4, the results on the benchmark problems are discussed and the performance of the proposed algorithms is compared. Finally, Section 5 concludes the work with a summarizing discussion.

## 2. Problem definition and preliminaries

### 2.1. Problem definition

In general, a multi-objective optimization problem is a problem with two or more objective functions to be optimized simultaneously. Such problems are defined as:

$$f_1(\mathbf{x}) \to \min, \cdots, f_m(\mathbf{x}) \to \min \tag{1}$$

subject to

$$g_1(\mathbf{x}) \leq 0, \cdots, g_k(\mathbf{x}) \leq 0 \tag{2}$$

$$\mathbf{x} \in \mathscr{X} \tag{3}$$

Here $x$ is an element in decision space $\mathscr{X}$, that is the space of all possible alternative solutions. $f_i(x)$ is the $i$th function of the $m$ objective functions and $g_i(x)$ is the $i$th function of the $k$ constraint functions. The search spaces could be continuous or discrete. In this article, the search space is a discrete one and the focus is on problems with three objective functions.

The multi-objective IRP with uncertain demand can be described as follows. In this problem, products are repeatedly delivered from a single supplier to a set of $n$ geographically dispersed customers over a given planning horizon $T$ (in days). On different days, each customer consumes a stochastic amount of the product. Moreover, customers maintain a local inventory with a maximum inventory level. The supplier has to service all customers with a fleet of homogeneous vehicles with an equal maximal capacity. The objective in [31] was to minimize the total inventory cost and the total routing cost during the planning period. Here, it is extended to a tri-objective problem that also states expected stockout cost as an objective function.

Formally, the problem is set up as described in [31]: There are $n$ customers and at most one vehicle with capacity $C$ per customer. Deliveries cannot be split in the model, namely a customer can be visited at most once by a vehicle per day. Each customer $i \in \{1, \cdots, n\}$ has a maximum inventory level denoted with $Q_i$. For each customer $i \in \{1, \cdots, n\}$ and time $t \in \{1, \cdots, T\}$ (denoting an index for the days), $L_{i,t}$ denotes the inventory level, $q_{i,t}$ is the shipping quantity, $d_{i,t}$ is the demand to be satisfied. At day 1 the values of $L_{i,t}$ are set to some pre-defined initial inventory level. The inventory levels are then updated according to the equation given by Geiger and Sevaux [31].

The stockout cost of this problem can be computed as $S_{i,t} = \max\{0, d_{i,t} - L_{i,t-1} - q_{i,t}\}$. A positive $S_{i,t}$ means that there are not enough products available at time $t$ for customer $i$. In [31], positive $S_{i,t}$ were avoided altogether by considering them as strict constraints. However, if demands are stochastic meaning they are not known beforehand, stockout cost cannot be avoided.

A solution candidate is represented by a tuple of delivery frequencies $(\pi_1, ..., \pi_n)$ with $\pi_i \in \{1, \cdots, T\}$ for $i = 1, \cdots, n$. For each customer it determines how often it is visited by a vehicle. For instance, $\pi_i = 1$ means a day-to-day delivery for customer $i$, $\pi_i = 2$ indicates that on every second day a delivery takes place, and so forth. The required shipping quantities are then determined by

$$q_{i,t} = \min\left\{ \left( \sum_{\ell=t}^{t-1+\pi_i} d_{i,\ell} \right) - L_{i,t-1},\ Q_i - L_{i,t-1},\ C \right\}$$

The two objectives which were stated in [31] are defined as:

$$f_1 = \sum_{t=1}^{T} \sum_{i=1}^{n} L_{i,t} \to \min \tag{4}$$

$$f_2 = \sum_{t=1}^{T} VRP_t(q_{1,t}, \ldots, q_{n,t}) \to \min \tag{5}$$

where Eq. 4 describes the total inventory cost and Eq. 5 represents the total cost for the routing. The latter is determined by solving for each day a vehicle routing problem $VRP_t(q_{1,t}, \cdots, q_{n,t})$ with the given shipping quantities $(q_{1,t}, \cdots, q_{n,t})$ for day $t$ (they are determined by the frequencies).

The vehicle routing problem is defined in the standard way: Given a set of $n$ customers and a depot, it is required to visit each customer exactly once and deliver the quantity $q_i$ to customer $i$. Multiple vehicles can be used and each vehicle has the same capacity. One vehicle must start from a depot and return to it. The number of vehicles is flexible, and therefore the constraints can always be satisfied. For a formal problem description, see [54]. The routing cost is proportional to the total distance of all tours.

Finally, stockout cost is defined by computing the total stockout in the scenario as:

$$f_3 = \sum_{t=1}^{T} \sum_{i=1}^{n} S_{i,t} \to \min \tag{6}$$

Therefore, this problem can be defined as a multi-objective optimization problem.

### 2.2. Multi-objective optimization

Miettinen [41] distinguishes the *a-priori* and *a-posteriori* methods for multi-objective optimization. In the *a-priori* approach, first an aggregating utility functions are defined and then the optimization is carried out.

In the *a-posteriori* approach a set of non-dominated solutions is computed and presented to the decision maker for selection and trade-off assessment. In this work, the *a-posteriori* approach is applied and algorithms are proposed to compute approximations to the non-dominated set (or Pareto front) of the multi-objective optimization problem.

In order to introduce a preorder on the search space, the usual Pareto dominance relation is used . A point $\mathbf{x}^{(1)} \in \mathscr{X}$ is said to *(Pareto) dominate* a point $\mathbf{x}^{(2)} \in \mathscr{X}$, if and only if for all objective function values $\mathbf{x}^{(1)}$ is not worse than $\mathbf{x}^{(2)}$, and $\mathbf{x}^{(1)}$ is strictly better than $\mathbf{x}^{(2)}$ in at least one objective function value. The set of non dominated solutions in $\mathscr{X}$ will be called the *efficient set* $\mathscr{X}^*$ and its image set $PF = \{\mathbf{f}(\mathbf{x}) | \ \mathbf{x} \in \mathscr{X}^*\}$ is commonly termed the *Pareto front*.

A performance indicator for how well a Pareto front is covered is the *hypervolume indicator* of a population. The hypervolume indicator defined in this way is a standard indicator in Pareto optimization and it has been shown by Zitzler et al. [59] that it has favorable properties compared to all other indicators when the goal is to obtain a well distributed approximation of the Pareto front. It is defined by the size of the dominated subspace in the objective space:

$$HV(\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}) = \lambda \left( \bigcup_{i=1}^{n} \left[ \mathbf{f}(\mathbf{x}^{(i)}), \mathbf{r} \right] \right) \quad (7)$$

Here $\lambda$ denotes the Lebesgue measure in dimension $m$. $\mathbf{r}$ is the reference point of this problem. It is assumed that the objective function vectors $\mathbf{f}(\mathbf{x}^{(i)})$ dominate $\mathbf{r}$, so that the orthogonal ranges $\left[ \mathbf{f}(\mathbf{x}^{(i)}), \mathbf{r} \right] \subset \mathbb{R}^m$ are well defined for $i = 1, \dots, n$.

Informally, the hypervolume indicator can be defined as a measure for the size of the set of dominated objective vectors, for which there exists a solution $\mathbf{x} \in \mathscr{X}$ that is either better or equal to the objective vector. To make this measure finite, the measured set is cut from above by a reference point. Increasing the hypervolume indicator means therefore to increase the amount of available options.

In Figure 1, it can be seen a projection of some population of 5 points to the objective space for some hypothetical function . The points $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, $\mathbf{x}^{(3)}$, and $\mathbf{x}^{(4)}$ are non-dominated and the point $\mathbf{x}^{(5)}$ is dominated. For each point the dominated part of the reference space is indicated by a gray shaded box surrounded by a dark gray line. The hypervolume indicator of this population and reference point $r$ equals the size of the entire gray shaded area. In this article algorithms are compared which seek to maximize the hypervolume indicator over the set of all subsets of $\mathscr{X}$ of size $n$. The algorithms produce a sequence of so called approximation sets that gradually converges to a diverse approximation of the Pareto front with maximal hypervolume indicator. Indicator-based multi-objective optimization (IMO) seek to improve the performance indicator of a solution set and by doing so achieve a good approximation to a Pareto front.

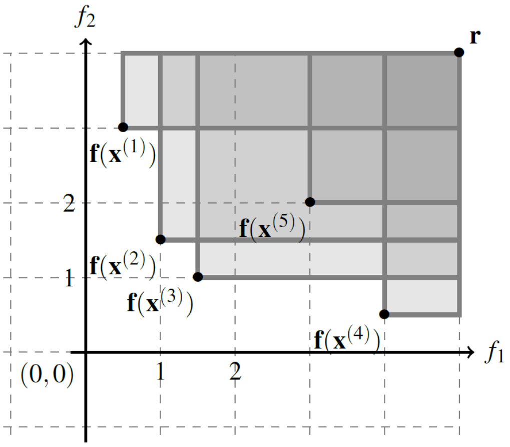
$f_2$
r
$\mathbf{f}(\mathbf{x}^{(1)})$
2
$\mathbf{f}(\mathbf{x}^{(5)})$
$\mathbf{f}(\mathbf{x}^{(2)})$
1
$\mathbf{f}(\mathbf{x}^{(3)})$
$\mathbf{f}(\mathbf{x}^{(4)})$
$f_1$
(0,0)
1
2

Fig. 1. Projection of a population to the objective space.

## 3. Multi-objective optimization algorithms

Next we will introduce the four algorithms that will be used for solving the three objective inventory routing problem. All algorithms are population-based metaheuristics, that is metaheuristics that maintain a population of search points. They aim to move the points closer to and across the Pareto front.

We will start with the introduction of the two IMO methods, namely the SMS-EMOA and MOCOPS. The SMS-EMOA is an evolutionary algorithm and the MOCOPS is a swarm based algorithm. In mathematical programming, evolutionary algorithms and particle swarm algorithms, different terminologies are used. An approximation set (to the efficient set) in mathematical programming is called a population of individuals in evolutionary algorithms and a swarm of particles in swarm algorithms. Iterations of an iterative search algorithm correspond to generations in evolutionary algo-

Table 1
Terminology in set-oriented optimization.

| Evolutionary Optimization | Swarm Optimization | Mathematical Programming |
|---|---|---|
| Population | Swarm | Approximation set |
| Individual | Particle | Decision vector |
| Generation | Position update | Iteration |

rithms and position updates in swarm algorithms. The terminology is summarized in Table 1.

In this article, both SMS-EMOA and MOCOPS aim to maximize the hypervolume indicator of a population of solutions and thereby create a diversified set on the Pareto front. SMS-EMOA is an evolutionary algorithm with single point replacement selection scheme (steady state selection), and MOCOPS is a swarm based algorithms where each point in the population is viewed as a search agent that seeks to improve its individual contribution to the hypervolume indicator. The structure of the solvers for inventory routing problem can be seen in Figure 2. The procedure starts with initialize the population. Based on the delivery frequencies given by the initialization, customers are assigned to be serviced at a certain day. For each day, how much quantity of products should be delivered would be decided which construct a classical vehicle routing problem. After the evaluation of one solution is done, the hypervolume contribution of each customer is calculated. Then, the population is updated according the hypervolume contribution of each customer. This loop is reiterated until a user defined termination criterion is reached.

### 3.1. *Initialization procedure*

In order to make the population evenly spread in the search space, a special initialization procedure is implemented which can be seen in [31]. First, solutions are initialized by given an identical delivery periods starting with 1 and increasing in step of 1 until no non-dominated alternatives can be added to the population anymore. For example, if there are 3 customers to be served with a max period of 3, then the initial delivery periods are $\pi$= { [1,1,1,1], [2,2,2,2], [3,3,3,3]}. The delivery periods are randomly set to $j$ or $j+1$, which can fill the gaps between the purely identical delivery period solutions. The initial delivery periods with size of 8 solutions would be $\pi$= { [1,1,1,1], [2,2,2,2], [3,3,3,3], [2,1,1,1], [2,2,2,1], [2,2,3,1], [3,3,2,2], [3,2,2,3]}. It will be decided based on the delivery frequencies, when and how many of the products should be delivered. For each period, a classical vehicle routing problem is defined.

Since the vehicle routing problem is an NP-hard problem [37,55], it is really time-consuming to solve this problem repeatedly. Therefore, the classical savings algorithm [18] is applied to construct of the solution of the vehicle routing problem. This algorithm is very fast and known to find good solutions on a wide range of problems. A number of heuristic approaches such as genetic algorithm and ant colony algorithm are able to improve the performance of VRP [24,28,30,51,52]. However, these algorithms are not chosen in this article because they are really time consuming.

### 3.2. *Evolutionary algorithm: SMS-EMOA*

In this SMS-EMOA points can be removed and new points might appear in the course of evolution [9,26]. Viewing as stochastic systems, it is a branching process, with 'birth events' creating a new branch, and 'death events' terminating a branch.

The SMS-EMOA is otherwise very similar to the swarm-based algorithm, as it bases the decisions on hypervolume contributions of points. Given a population $P$, the hypervolume contribution $\Delta H(\mathbf{x})$ is defined as: $\Delta HV(\mathbf{x}) = HV(P) - HV(P \setminus \{\mathbf{x}\})$

For visualization of a population with 3-D objective vectors, see Figure 3. An asymptotically optimal algorithm for computing all hypervolume contributions in a population of 3-D vectors has been discovered by Emmerich and Fonseca [27]. The running time complexity is $O(|P|log|P|)$, here $|P|$ is the size of Pareto front set, and this step is therefore up to a constant factor as fast as computing the hypervolume of a population.

The simple version of SMS-EMOA that was used in our experiments is outlined in Algorithm 1. After initializing a population, in each iteration, first a new solution is created by mutating an existing solution *(Mutate(.))*. Then it is added to the population. Subsequently, the hypervolume contributions of all population members are computed and those with the least contribution are determined. A randomly chosen 'least hypervolume contributor' is then discarded from the population. Due to the last step the population size is kept constant and there is a selection pres-sure towards sets which cover more hypervolume.

### 3.3. *Multi-objective cooperative particle swarm*

As opposed to the evolutionary algorithm, multi-objective optimization cooperative particle swarms

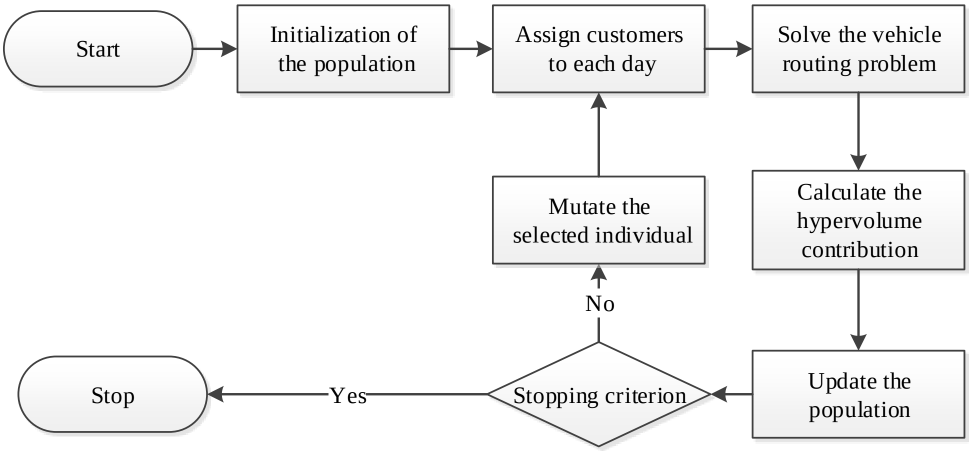
Start
Initialization of the population
Assign customers to each day
Solve the vehicle routing problem
Calculate the hypervolume contribution
Update the population
Stopping criterion
Mutate the selected individual
No
Yes
Stop

Fig. 2. The structure of multi-objective optimization algorithm for inventory routing problem

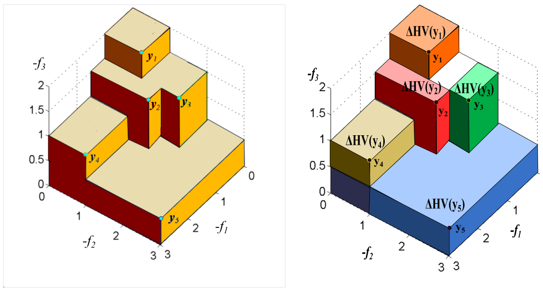

Fig. 3. Left: Hypervolume dominated by a set of 3-D objective vectors. Right: *Hypervolume contributions* of these vectors.

**Algorithm 1** S-Metric Selection Evolutionary Algorithm (SMS-EMOA)*

```
Input initial population $P_0$
while termination criterion is not reached do
    $t \leftarrow t+1$
    $x^{(s)}$: Random select on individual from the population
    $\mathbf{x}_{old} = \mathbf{x}^{(s)}$
    $\mathbf{x}_{new} = Mutate(\mathbf{x}_{old})$
    $P \leftarrow P_t \cup \{\mathbf{x}_{new}\}$
    {Determine the (set of) least hypervolume contributors}
    $\mathscr{L} \leftarrow \arg\min_{\mathbf{x} \in P} \Delta HV(\mathbf{x})$
    Chose randomly a solution $x^{(s)}$ in $\mathscr{L}$
    $P \leftarrow P_t \setminus \{\mathbf{x}^{(s)}\}$
end while
```

(*Simple version with random selection among dominated solutions.)

(MOCOPS) is a randomized search heuristics where a swarm of particles moves gradually towards a hypervolume-maximizing solution set driven by randomized modification operators and interaction between the particles. It could be described as a multi-trajectory stochastic flow. It is related to particle swarm optimization algorithms (PSO) that have been suggested for other search problems in the literature [10,47].

In conventional PSO algorithms, the swarm is driven by a leader, who is the currently best individual in a population, and by local memories of particles of their so-far best positions. In single-objective optimization such processes will typically converge to local, or sometimes even to global optima. In multi-objective optimization such an approach could be easily used to find a single point on the Pareto front, but is not well suited to distribute points across the Pareto front, because the particles all strive to resemble the leader which is counter-productive when searching for a diverse set of solutions. To a certain extent this can be compensated by assigning local leaders, but this makes the algorithm quite complicated and adds parameters to the algorithm (i.e., number of leaders).

In multi-objective optimization there is no definition of a best solution and thus there is no obvious choice for a leader individual. However, one could for instance form subpopulations or use local metrics. The use of traditional PSO for multi-objective opti-mization problems has been addressed already in the literature, both in the context of general multi-objective optimization [22] and for finding Pareto fronts that maximize the hypervolume indicator [43,44,45]. Both approaches lead to algorithms pro-duce good approximations to Pareto fronts. The ver-sion of multi-objective particle swarm optimization used in this article resembles closely [43], but it is using a simplified leader free selection and variation scheme.

For the cooperative particle swarm algorithm, the following properties distinguish it from previous swarm-based approaches:

- Leader-free: The particles in the population cooperate in covering the Pareto front, instead of competing with each other. There is thus no leader in the swarm; each particle strives to contribute to the global performance of the swarm.
- Indicator-based: The algorithm seeks to maximize a unary performance indicator. Here the hypervolume indicator is used.

The new approach is deliberately kept very simple. This is for two reasons: Firstly we want to demonstrate that only a few essential components are needed to steer a swarm towards a Pareto front. Secondly, simplicity will make the algorithms easier accessible to a rigorous theoretical analysis.

Here we term this approach multi-objective optimization cooperative particle swarms. The particles in the swarm strive to contribute as much as possible to the team performance of the population. In doing so, they seek to contribute in different ways to the goal of covering the Pareto front.

In the traditional PSO algorithms a swarm will thus lead to unity, whereas in multi-objective optimization you need diversification to adequately approach the Pareto set. In trivial situations where the objectives are not conflicting, no diversification is needed. In this case, the problem can easily be restated as single-objective problems instead. The pseudo-code for the proposed MOCOPS algorithm is given in Algorithm 2. It starts with randomly initializing a set of particles. Then, in each iteration of the algorithm, a particle is randomly selected and a small random variation of this particle is generated by adding a random perturbation (Mutate(.)).

If the fitness contribution of the mutated particle relative to the population is better than for the original position then the particle will move to the new position: Firstly, it will be tested which one of the two positions leads to a better hypervolume indicator of the population. Secondly, if both positions are equally good (which will typically occur for dominated solutions), the point that has a better value in the aggregated linear objective function with equal weights is considered. Note that if one solution is dominated by the other solution it will also be considered better in the latter comparison (because of positive equal weighting). Therefore, eventually all solutions will strive towards the non-dominated front and then their hypervolume contribution will be considered. The cycle continues with picking a random particle again. Care must be taken to ensure $\vec{\mathbf{x}}_{new} \in \mathbb{S}$ (e.g. by rejecting infeasible vectors).

**Algorithm 2** Multi-objective Optimization by Cooperative Swarms (MOCOPS)

```
Input initial population $P_0$
while termination criterion is not reached do
    $t \leftarrow t+1$
    $x^{(s)}$: Random select on individual from the population
    $\mathbf{x}_{old} = \mathbf{x}^{(s)}$
    $P \rightarrow P_t \setminus \{\mathbf{x}^{(s)}\}$
    {Try to improve position of particle $\mathbf{x}^{(s)}$}
    $\mathbf{x}_{new} = Mutate(\mathbf{x}_{old})$
    if $HV(P \cup \{\mathbf{x}_{new}\}) > HV(P \cup \{\mathbf{x}_{old}\})$ then
        $P_t = P \cup \{\mathbf{x}_{new}\}$
    else if $HV(P \cup \{\mathbf{x}_{new}\}) < HV(P \cup \{\mathbf{x}_{old}\})$ then
        $P_t = P \cup \{\mathbf{x}_{old}\}$
    else if $f_1(\mathbf{x}_{new}) + f_2(\mathbf{x}_{new}) < f_1(\mathbf{x}_{old}) + f_2(\mathbf{x}_{old})$ then
        $P_t = P \cup \{\mathbf{x}_{new}\}$
    else
        $P_t = P \cup \{\mathbf{x}_{old}\}$
    end if
end while
Return $P_t$
```

One iteration of the bi-objective MOCOPS algorithm can be performed with a time complexity in $O(|P|log|P|)$. Here $|P|$ is the size of the Pareto front set. This can be achieved by using a dimension sweep algorithm and an AVL tree [27]. However, by implementing the algorithm as an online algorithm, that is using incremental update steps, we can compute a single iteration with time complexity in $O(log|P|)$ (amortized over the number of iterations) [35]. This algorithm dynamically updates the AVL tree keeping non-dominated points sorted in the first coordinate. Fast - linear time - hypervolume update schemes are also known for three objective functions [32]. The computational complexity is expected to grow exponentially in the number of objective functions [11]. For this reason the scheme does probably not lend itself very well for many-objective optimization.

On first glance MOCOPS and SMS-EMOA look similar. However, there is an important difference. The difference is given by the stochastic dynamics of the two algorithms. The SMS-EMOA is a branching process - a single (parent) point can generate multiple offspring over time or it might also disappear without ever producing an offspring. In MOCOPS a point is either preserved or replaced by a better neighboring point (local move). In this sense the dynamics of MOCOPS is simpler, but it is less likely in MOCOPS to abandon subspaces entirely, as it might occur in SMS-EMOA.

### *3.4. NSGA-II*

NSGA-II is a classical multi-objective algorithm proposed by Deb et al. [23]. In each iteration, the tournament selection, recombination and mutation operator are used to generate the offspring. Then a fast nondominated sorting method and crowding distance as a secondary ranking order are applied to select the population for the next generation which can maintain the diversity of the population. Since in the IRP the solution is an integer vector, the operators are adapted to a perturbation in order to implement NSGA-II.

### *3.5. HGS14*

In addition, we extended the state-of-the-art decomposition based bi-objective optimization algorithm for bi-objective inventory routing by Huber, Geiger, and Sevaux (HGS14) to the 3-D case [34]. They use the same initialization procedure as described in this paper. All nondominated initial solutions are added to an archive. Then, in the improvement procedure, a set of reference points are selected. To select the representatives of solutions, the objectives of solutions are normalized and the solutions that minimize the distance to these reference points are selected from the archive in each iteration. For each reference point the nearest neighbor in the Chebychev distance is selected from the archive and then improved by local search, also minimizing this Chebychev distance. New solutions will be added to the archive and the dominated solutions in the archive are deleted. Then new reference points are selected are determined based on the new nondominated set, and the procedure is repeated. The process is terminated after a prescribed number of steps. The local searches are exhaustive and use as local search operator increment or decrement of a single vector position by 1.

A main extension is the reference points selection methods from 2-D to 3-D. A regular pattern is used to distribute the reference point on the boundaries of a simplex which dominates the current Pareto front. Suppose the space in each dimension is divided into $n$ intervals. Then the value of each dimension the reference points should belong to the set $s = \{s_i | s_i = i/n, i = 0, \cdots, n\}$. Then the reference points set can be defined as $R = \{\overrightarrow{r} | \forall i \in \{1,2,3\} : r_i \in s \wedge \exists j \in \{1,2,3\} : r_j = 0\}$. For instance, for $n = 3$, we construct the reference points by the base pattern {(0,0,0); (0,0,1/3); (0,0,2/3); (0,0,1); (0,1/3,0); (0,1/3,1/3); (0,1/3,2/3); (0,2/3,0); (0,2/3,1/3); (0,1,0); (1/3,0,0); (1/3,0,1/3); (1/3,0,2/3); (1/3,1/3,0); (1/3,2/3,0); (2/3,0,0); (2/3,0,1/3); (2/3,1/3,0); (1,0,0)}. This example of the reference points can be seen in Figure 4 and is used in the studies for this paper. The base pattern is rescaled by multiplying each component of a base-vector by the maximal value for that dimension in the current Pareto front approximation. Note that, NSGA-III also use the concept of reference points. However, they use the reference plane to select them [50].

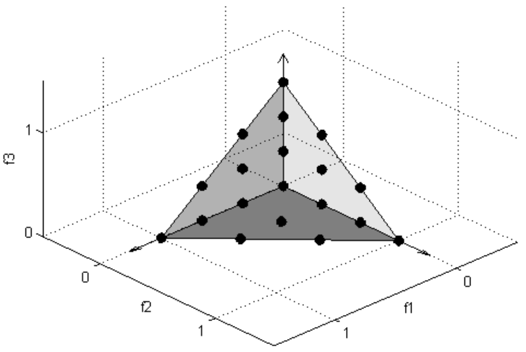

Fig. 4. An example of refenence points selected on 3-D method

## 4. Computing results

### 4.1. Test instances

The test benchmark instances are proposed by Sevaux et al. [31], which are available from `http://logistik.hsu-hh.de/IRP`. The concept of multi-period demand is added to the classical VRP benchmark data. They generated a total of $T = 30$ periods demand of all customers. In each period, the given demand range from $-25\%$ to $25\%$ around the average demand. The number of customers for problem GS-01, GS-02, GS-03, GS-04 and GS-05 are 55, 75, 100, 150 and 200, respectively. In this article, an extension is made to generate the uncertain demand. The period demand of each customer is given as a random number with a Poisson distribution whose expected value is $\mu$, the average period demand which is given in Sevaux's benchmark. For a typical value of $\mu$, that is $\mu = 20$, the probability density function of this Poisson distribution can be seen in Figure 5.

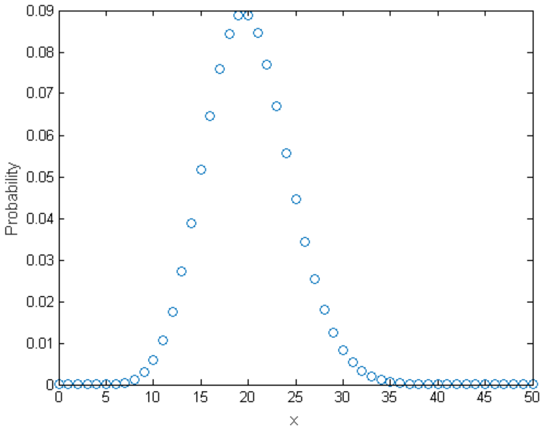

Fig. 5. The specific probability density function of the demand

### 4.2. Parameter settings

All four algorithms are tested on the bi-objective and tri-objective inventory routing problem. The matlab code is made available via `http://natcomp.liacs.nl/index.php?page=code`. The algorithm setup was as follows: For the mutation integer mutation with geometrical distribution is used [39] and if interval boundaries are exceeded, they are set to boundary. The vehicle routing was done with the parameter-free savings heuristics. All other problem data was chosen according to [31]. The runs were conducted for 5000 evaluations of the objective functions.

Table 2
The reference points used in different instances

| Instances | reference points |
|---|---|
| GS-01 | (233100,72070.43,20895) |
| GS-02 | (409200,108925.70,36949) |
| GS-03 | (437400,149682.68,39494) |
| GS-04 | (670500,220815.05,60500) |
| GS-05 | (955800,288252.24,86207) |

For the hypervolume indicator, the reference point $R = (r_1, r_2, r_3)$ was used. In order to determine a reference point, an upper bound for all objective function values was required: For the first objective (inventory cost) $r_1$, the assumption is that all inventories are always at their maximum allowed level. For the second objective (routing cost) $r_2$, the assumption is in every day each customer is severed with one vehicle. Finally, for the third objective (stockout cost) $r_3$, the assumption is no deliveries would take place. The reference points used in different test instances can be seen in Table 2.

For the population based algorithm, population size is a sensitive parameter that should be carefully treated. In this article, the parameter tuning is done by changing the population size from 10 to 100 with an interval of 10. In Figure 6, it can be seen that the hypervolume values increase with the population before 80 and then drop gradually. The results show that the best population size should be set to 80. Note that large populations cause longer computing time, due to resource sharing.

### 4.3. Biobjective IRP experiment

The first comparison is on the bi-objective problem. Figure 7 shows the result of the comparison. Clearly the results of both algorithms seem to be very similar. The most left point appears as an isolated solution around $(0.5, 1.6)$. The delivery frequencies for all customers are 1, which means every day all customers are served. Moving further right by holding more inventories would make a big reduction of routing cost. However, after inventory cost reaching 5, the inventory cost would not increase so much any more, because maximum inventory levels are reached. The results also show that NSGA-II, SMS-EMOA, MOCOPS can get similar results in bi-objective inventory routing problem. HGS14 per-forms a little better in the problem.

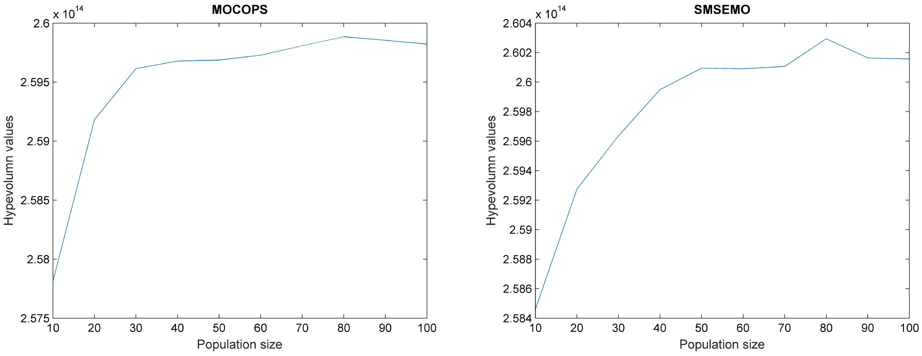

Fig. 6. Left: The hypervolume indicator of MOCOPS changes with population sizes. Right: The hypervolume indicator of SMS-EMOA changes with population sizes.

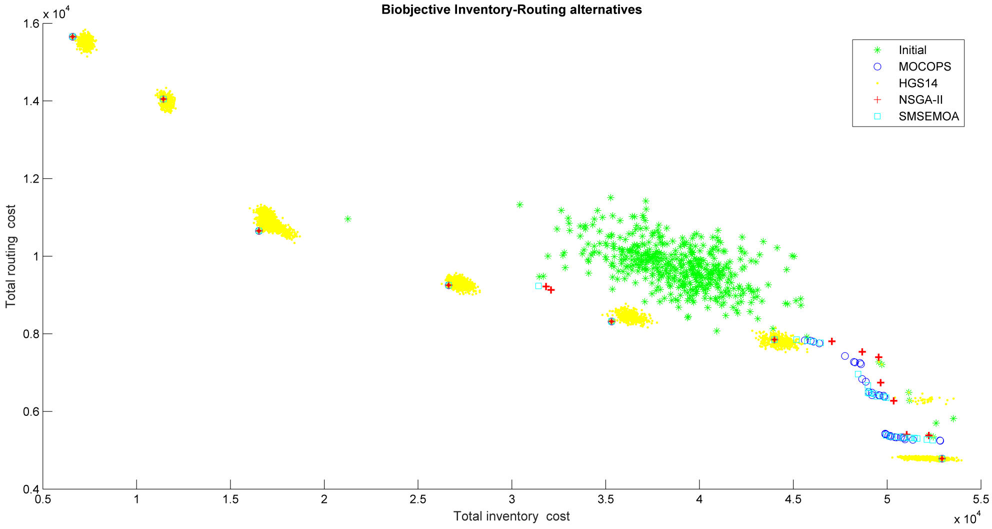

Fig. 7. Pareto front approximations of the bicriteria problem obtained with different algorithms.

### 4.4. *IRP with uncertain demand experiments*

In this experiment, all four algorithms are executed 10 times on 5 test instances with the number of customers varying from 50 to 200. Since Sevaux et al. only provided results of bi-objective IRP, there is no existing results that can be used as a comparison on this problem. Therefore, these two algorithms are compared to each other, which provide some preliminary insights into the Pareto optimal front of the IRP with uncertain demand.

In order to visualize Pareto front approximations in the 3-D case, attainment surface plots are used. The attainment-surface separates the dominated subspace (grey volume) from the non-dominated subspace. Using the same notation than in Section 3.2 it can be defined as the set of points that are only weakly dominated by some points in Pareto front set $P$, i.e. $\{y|\exists y' \in P : y' \leq y \wedge \neg \exists y'' \in P : y'' < y\}$, where '$\leq$' denotes the weak componentwise order and '$<$' denotes the strict componentwise order. In order to more accurately assess the quality of single points we also provided the three projections as a scatter plot. An example of 3D visualization of Pareto-front generated by these two algorithms on problem GS-01 can be seen in Figure 8 and Figure 9. Other results are shown in Figure 10 and Figure 11.

As we can see, SMS-EMOA and MOCOPS produced very similar results, which could be interpreted as an indication that a good approximation to the hypervolume maximal front was obtained. The interpretation of 3-D results is more involved, as three trade-offs need to be taken into account: Firstly, from the projection to Routing cost and Inventory cost we obtain a similar set of non-dominated solutions, in the 2-D projection, to the 2-D study, which means there is a clear conflict between the inventory cost and routing cost. Then, from the projection of routing cost and stockout cost, it can be seen that there is also a correlation between them. In order to decrease the stockout cost, the decision makers could try to make the delivery frequency smaller, which would increase the routing cost. The stockout cost would reach the optima at the point where the routing cost is around 10000. The maximal inventory levels are often reached in this case. The relation between stockout cost and inventory cost is special. It is either possible to improve stockout cost by increasing the routing cost, or by increasing the inventory cost. For very low high values of the $\pi$ vector, routing costs will be low but the demands of customers cannot be satisfied. This will eventually increase the stockout cost. There is also a weak decrease of the stockout cost with growing inventory costs before the inventory reaches a value of ca. 30000. Then the stockout cost grows rapidly with increasing inventory cost. These points have a high inventory cost and a high stockout cost. The reason why they are still non-dominated is that they have a small routing cost. The infrequent delivery, causes the raise of stockout costs.

In order to compare the quality of the Pareto front approximations of these algorithms, the hypervolume of all test instances are given in Table 3. The results show that these four algorithms could get a similar hypervolume for all test instances. HGS14 works better in small scale instances such as GS-01 and GS-02. SMS-EMOA has a good performance on GS-03. However, the MOCOPS could achieve a better performance in large scale instance such as GS-04 and GS-05. The NSGA-II seems to run faster than other algorithms in most cases. MOCOPS runs faster on GS-02 and GS-03.

## 5. Conclusion and outlook

In this article the bi-objective inventory routing problem is extended to a tri-objective inventory rout-ing problem by introducing stockout cost as a third objective function. This is important in the context of uncertain demand distributions. It was argued that a Poisson random variable can be used to model the demand and resulting stockout costs.

The paper studied four population based metaheuristics to compute the 3-D Pareto front of the tri-objective inventory routing problem. Two indicator based algorithms for multi-objective optimization were applied to compute the Pareto fronts, namely NSGA-II, HGS14, SMS-EMOA and MOCOPS. The NSGA-II and SMS-EMOA are state-of-the-art evolutionary optimization method, whereas the MOCOPS was a new, customized version of a swarm based optimizer. It was shown that, if time for optimization is limited, the population size is a crucial control variable and values around a size of 80 achieve a good performance in realistic settings. All indicator based algorithms achieved similar results in various runs and this consensus makes us belief that they were both able to find near optimal approximation sets to the Pareto front. On small instances, HGS14 per-formed best and as big instances MOCOPS.

The resulting Pareto front revealed interesting insights into the trade-off between different objectives. The results confirm that the stockout cost is in conflict with the distance cost and inventory cost. To reduce stockout cost one has to either accept higher inventory costs or higher distance costs. This is ob-served across several benchmark instances and the 3-D Pareto fronts have a similar shape for the different benchmark problems. Hence, we can conclude that there is a something like a typical (parabolic) Pareto front shape for a tri-objective vehicle routing problem, and it is possible to compute these Pareto fronts in practically feasible time. Computation times range from 20 minutes (50 customers) to 8 hours (200 customers) on a typical desktop PC.

In the future to allow for real-time optimization, it will be interesting to further reduce computation time for large instances, for instance by using more efficient and precise search procedures. Moreover, more advanced algorithms should be implemented to solve the subordinated vehicle routing problem, such as ant colony algorithm and genetic algorithm. Finally, an interesting endeavor will be the study of real world applications with empirically fitted demand distributions including pilot experiments.

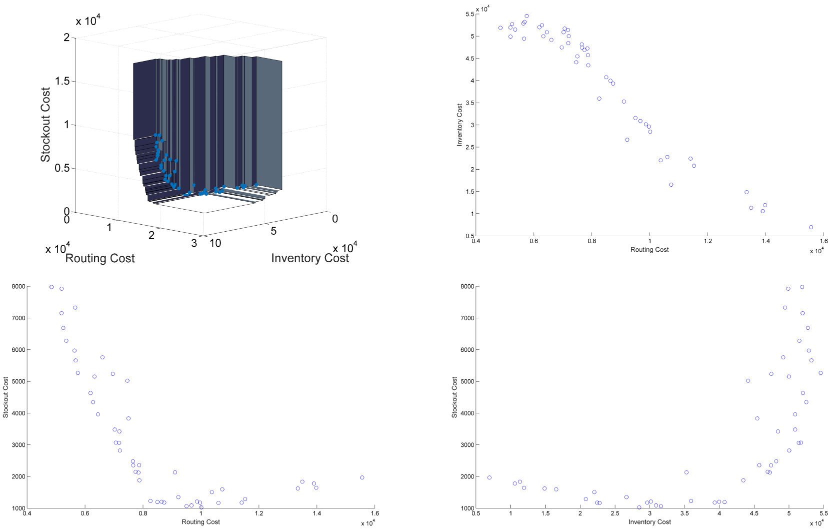

Fig. 8. Plots of tricriteria MOCOPS for IRP with uncertain demands : Points and dominated subspace (upper left), inventory cost vs. routing cost (upper right), stockout cost vs. routing cost (lower left),stockout cost vs. inventory cost (lower right).

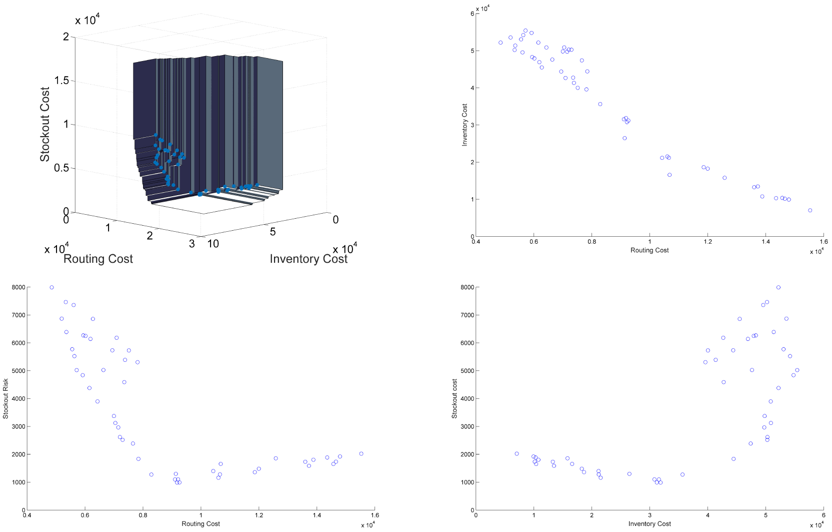

Fig. 9. Plots of tricriteria SMS-EMOA for IRP with uncertain demands: Points and dominated subspace (upper left), inventory cost vs. routing cost (upper right), stockout cost vs. routing cost (lower left),stockout cost vs. inventory cost (lower right).

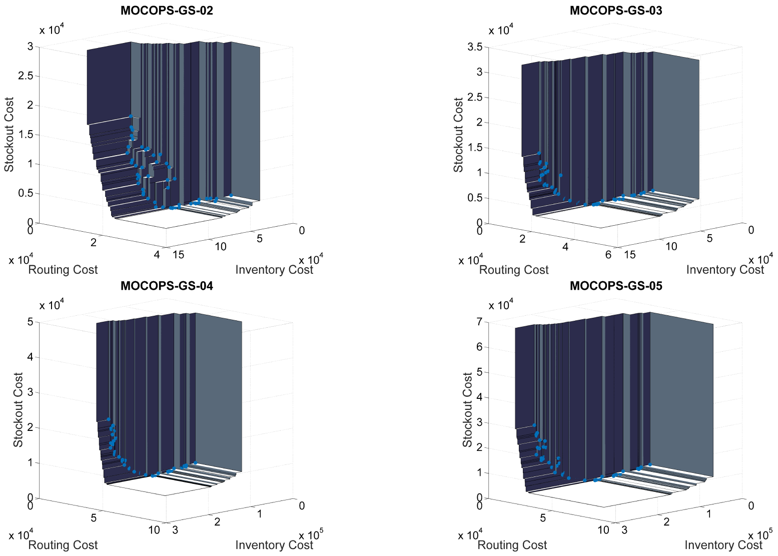

Fig. 10. 3D Plots of tricriteria MOCOPS for IRP with uncertain demands

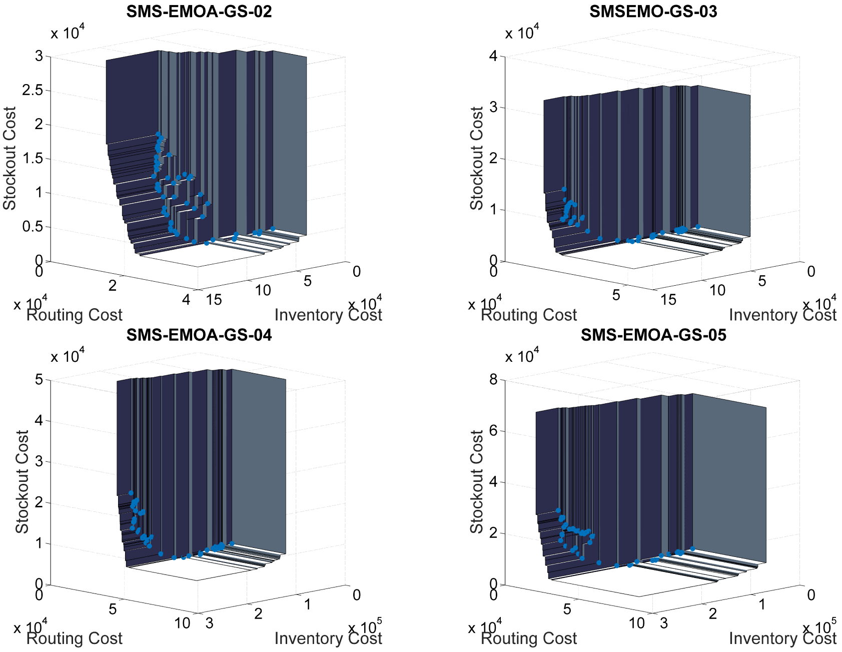

Fig. 11. 3D Plots of tricriteria SMS-EMOA for IRP with uncertain demands

## Acknowledgement

The authors gratefully acknowledge financial support by Agentschap NL, The Netherlands within the project ‘Deliver’. Zhiwei Yang gratefully acknowledges the financial support from China Scholar Council (CSC), CSC No.201206110020. A conference version containing part of this paper appeared under the

Table 3
Hypervolume values and computation time of all problems

| | | NSGA-II | | | HGS2014 | | | SMS-EMOA | | | MOCOPOS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Test | Pop | HV | Std | CT | HV | Std | CT | HV | Std | CT | HV | Std | CT |
| Case | Size | $(10^{11})$ | $(10^{11})$ | (h) | $(10^{11})$ | $(10^{11})$ | (h) | $(10^{11})$ | $(10^{11})$ | (h) | $(10^{11})$ | $(10^{11})$ | (h) |
| GS-01 | 50 | 2932.14 | 5.67 | **0.7** | **2953.12** | 5.21 | 1.2 | 2952.57 | **3.60** | 0.8 | 2948.96 | 4.22 | 0.8 |
| GS-02 | 75 | 13640.85 | 27.25 | 1.4 | **13747.60** | 16.78 | 2.1 | 13694.38 | 21.58 | 1.5 | 13728.82 | **11.55** | **1.3** |
| GS-03 | 100 | 22014.79 | **22.42** | 2.2 | 22088.44 | 25.91 | 3.5 | **22093.473** | 23.199 | 2.6 | 22079.416 | 25.695 | **2.1** |
| GS-04 | 150 | 76362.16 | 152.40 | **4.2** | 76691.19 | **47.17** | 6.8 | 76587.74 | 112.64 | 5.0 | **76709.02** | 77.63 | 4.5 |
| GS-05 | 200 | 202419.1 | 182.13 | **6.8** | 203044.0 | **31.97** | 9.9 | 202971.27 | 209.33 | 8.1 | **203093.22** | 122.26 | 7.3 |

title "Multicriteria Inventory Routing by Cooerative Swarms and Evolutionary Algorithms" in the conference IWINAC2015 [56]. It contained less benchmark problems and only two out of four algorithms.

## References

[1] Y. Adulyasak, J.-F. Cordeau, and R. Jans. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. INFORMS Journal on Computing, 26(1):103–120, 2013.

[2] E.-H. Aghezzaf, B. Raa, and H. Van Landeghem. Modeling inventory routing problems in supply chains of high consumption products. European Journal of Operational Research, 169(3):1048–1063, 2006.

[3] S. Anily and A. Federgruen. One warehouse multiple retailer systems with vehicle routing costs. Management Science, 36(1):92–114, 1990.

[4] C. Archetti, L. Bertazzi, A. Hertz, and M. G. Speranza. A hybrid heuristic for an inventory routing problem. INFORMS Journal on Computing, 24(1):101–116, 2012.

[5] C. Archetti, L. Bertazzi, G. Laporte, and M. G. Speranza. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transportation Science, 41(3):382–391, 2007.

[6] D. Barnes-Schuster and Y. Bassok. Direct shipping and the dynamic single-depot/multi-retailer inventory system. European Journal of Operational Research, 101(3):509–518, 1997.

[7] W. J. Bell, L. M. Dalberto, M. L. Fisher, A. J. Greenfield, R. Jaikumar, P. Kedia, R. G. Mack, and P. J. Prutzman. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. Interfaces, 13(6):4–23, 1983.

[8] L. Bertazzi, G. Paletta, and M. G. Speranza. Deterministic order-up-to level policies in an inventory routing problem. Transportation Science, 36(1):119–132, 2002.

[9] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research, 181(3):1653–1669, 2007.

[10] I. Boulkaibet, L. Mthembu, F. De Lima Neto, and T. Marwala. Finite element model updating using fish school search and volitive particle swarm optimization. Integrated Computer-Aided Engineering, 22(4):361–376, 2015.

[11] K. Bringmann and T. Friedrich. Approximating the least hypervolume contributor: Np-hard in general, but fast in practice. In Evolutionary Multi-Criterion Optimization, pages 6–20. Springer, 2009.

[12] L. D. Burns, R. W. Hall, D. E. Blumenfeld, and C. F. Daganzo. Distribution strategies that minimize transportation and inventory costs. Operations Research, 33(3):469–490, 1985.

[13] A. Campbell, L. Clarke, A. Kleywegt, and M. Savelsbergh. The inventory routing problem. In Fleet management and logistics, pages 95–113. Springer, 1998.

[14] B. R. Campomanes-Álvareza, O. Cordón, and S. Damasa. Evolutionary multi-objective optimization for mesh simplification of 3d open models. Integrated Computer-Aided Engineering, 20(4):375–390, 2013.

[15] M. W. Carter, J. M. Farvolden, G. Laporte, and J. Xu. Solving an integrated logistics problem arising in grocery distribution. INFOR, 34(4):290, 1996.

[16] J. Cheng, G. Zhang, F. Caraffini, and F. Neri. Multicriteria adaptive differential evolution for global numerical optimization. Integrated Computer-Aided Engineering, 22(2):103–107, 2015.

[17] T. W. Chien, A. Balakrishnan, and R. T. Wong. An integrated inventory allocation and vehicle routing problem. Transportation Science, 23(2):67–76, 1989.

[18] G. Clarke and J. Wright. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12(4):568–581, 1964.

[19] L. C. Coelho, J.-F. Cordeau, and G. Laporte. Thirty years of inventory routing. Transportation Science, 48(1):1–19, 2013.

[20] L. C. Coelho and G. Laporte. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. International Journal of Production Research, 51(23-24):7156–7169, 2013.

[21] L. C. Coelho and G. Laporte. The exact solution of several classes of inventory-routing problems. Computers & Operations Research, 40(2):558–565, 2013.

[22] C. A. Coello Coello and M. S. Lechuga. Mopso: A proposal for multiple objective particle swarm optimization. In Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on, volume 2, pages 1051–1056. IEEE, 2002.

[23] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. Evolutionary Computation, IEEE Transactions on, 6(2):182–197, 2002.

[24] M. Dorigo and M. Birattari. Ant colony optimization. In Encyclopedia of machine learning, pages 36–39. Springer,

2010.

[25] M. Dror and M. Ball. Inventory routing: Reduction from an annual to short period problem. Université de Montréal, Centre de recherche sur les transports, 1984.

[26] M. Emmerich, N. Beume, and B. Naujoks. An emo algorithm using the hypervolume measure as selection criterion. In Evolutionary Multi-Criterion Optimization, pages 62–76. Springer, 2005.

[27] M. T. Emmerich and C. M. Fonseca. Computing hypervolume contributions in low dimensions: Asymptotically optimal algorithm and complexity results. In Evolutionary Multi-Criterion Optimization, pages 121–135. Springer, 2011.

[28] E. Forcael, V. González, F. Orozco, S. Vargas, A. Pantoja, and P. Moscoso. Ant colony optimization model for tsunamis evacuation routes. Computer-Aided Civil and Infrastructure Engineering, 29(10):723–737, 2014.

[29] G. Gallego and D. Simchi-Levi. On the effectiveness of direct shipping strategy for the one-warehouse multi-retailer r-systems. Management Science, 36(2):240–243, 1990.

[30] L. Gambardella, E. Taillard, and G. Agazzi. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In New Ideas in Optimization, chapter 5, pages 63–76. McGraw-Hill, 1999.

[31] M. J. Geiger and M. Sevaux. The biobjective inventory routing problem–problem solution and decision support. In Network Optimization, pages 365–378. Springer, 2011.

[32] A. P. Guerreiro, C. M. Fonseca, and M. T. Emmerich. A fast dimension-sweep algorithm for the hypervolume indicator in four dimensions. In CCCG, pages 77–82, 2012.

[33] S.-H. Huang and P.-C. Lin. A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty. Transportation Research Part E: Logistics and Transportation Review, 46(5):598–611, 2010.

[34] S. Huber, M. J. Geiger, and M. Sevaux. Interactive reference point-based guided local search for the bi-objective inventory routing problem. arXiv preprint arXiv:1405.5643, 2014.

[35] I. Hupkens and M. Emmerich. Logarithmic-time updates in sms-emoa and hypervolume-based archiving. In EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV, pages 155–169. Springer, 2013.

[36] L. Jia, Y. Wang, and L. Fan. Multiobjective bilevel optimization for production-distribution planning problems using hybrid genetic algorithm. Integrated Computer-Aided Engineering, 21(1):77–90, 2014.

[37] T. Kimura and T. Ikeguchi. An optimum strategy for dynamic and stochastic packet routing problems by chaotic neurodynamics. Integrated Computer-Aided Engineering, 14(4):307–322, 2007.

[38] A. J. Kleywegt, V. S. Nori, and M. W. Savelsbergh. The stochastic inventory routing problem with direct deliveries. Transportation Science, 36(1):94–118, 2002.

[39] R. Li, M. Emmerich, J. Eggermont, and E. G. Bovenkamp. Mixed-integer optimization of coronary vessel image analysis using evolution strategies. In Proceedings of the 8th annual conference on Genetic and evolutionary computation, pages 1645–1652. ACM, 2006.

[40] X. Li, W. Li, X. Cai, and F. He. A hybrid optimization approach for sustainable process planning and scheduling. Integrated Computer-Aided Engineering, 22(4):311–326, 2015.

[41] K. Miettinen. Nonlinear multiobjective optimization, volume 12. Springer Science & Business Media, 1999.

[42] N. H. Moin, S. Salhi, and N. Aziz. An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. International Journal of Production Economics, 133(1):334–343, 2011.

[43] S. Mostaghim, J. Branke, and H. Schmeck. Multi-objective particle swarm optimization on computer grids. In Proceedings of the 9th annual conference on Genetic and evolutionary computation, pages 869–875. ACM, 2007.

[44] S. Mostaghim, J. Branke, and H. Schmeck. Multi-objective particle swarm optimization on computer grids. In Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, GECCO '07, pages 869–875, New York, NY, USA, 2007. ACM.

[45] S. Mostaghim and J. Teich. Strategies for finding good local guides in multi-objective particle swarm optimization (mopso). In Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE, pages 26–33. IEEE, 2003.

[46] R. Ribeiro and H. Ramalhinho-Lourenço. Inventory-routing model, for a multi-period problem with stochastic and deterministic demand. 2003.

[47] A. Rodríguez and J. A. Reggia. A distributed learning algorithm for particle systems. Integrated Computer-Aided Engineering, 16(1):1–20, 2009.

[48] S. Rokni and A. R. Fayek. A multi-criteria optimization framework for industrial shop scheduling using fuzzy set theory. Integrated Computer-Aided Engineering, 17(3):175–196, 2010.

[49] M. Savelsbergh and J.-H. Song. An optimization algorithm for the inventory routing problem with continuous moves. Computers & operations research, 35(7):2266–2282, 2008.

[50] H. Seada and K. Deb. U-nsga-iii: A unified evolutionary optimization procedure for single, multiple, and many objectives: Proof-of-principle results. In Evolutionary Multi-Criterion Optimization, pages 34–49. Springer, 2015.

[51] N. Siddique and H. Adeli. Computational intelligence: synergies of fuzzy logic, neural networks and evolutionary computing. John Wiley & Sons, 2013.

[52] M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research, 35(2):254–265, 1987.

[53] O. Solyali and H. Süral. A branch-and-cut algorithm using a strong formulation and an a priori tour-based heuristic for an inventory-routing problem. Transportation Science, 45(3):335–345, 2011.

[54] B. van Veen, M. Emmerich, Z. Yang, T. Bäck, and J. Kok. Ant colony algorithms for the dynamic vehicle routing problem with time windows. In Natural and Artificial Computation in Engineering and Medical Applications, pages 1–10. Springer, 2013.

[55] Z. Yang, M. Emmerich, and T. Back. Ant based solver for dynamic vehicle routing problem with time windows and multiple priorities. In Evolutionary Computation (CEC), 2015 IEEE Congress on, pages 2813–2819. IEEE, 2015.

[56] Z. Yang, M. Emmerich, T. Bäck, and J. Kok. Multicriteria inventory routing by cooperative swarms and evolutionary algorithms. In Bioinspired Computation in Artificial Systems, pages 127–137. Springer, 2015.

[57] Q.-H. Zhao, S. Chen, and C.-X. Zang. Model and algorithm for inventory/routing decision in a three-echelon logistics system. European Journal of Operational Research, 191(3):623–635, 2008.

[58] Z. Zhu, J. Xiao, J.-Q. Li, F. Wang, and Q. Zhang. Global path planning of wheeled robots using multi-objective memetic algorithms. Integrated Computer-Aided Engineering, 22(4):387–404, 2015.

[59] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. Evolutionary Computation, IEEE Transactions on, 7(2):117–132, 2003.