Novel NSGA-II and SPEA2 Algorithms for Bi-Objective Inventory Optimization

Ilham HUSEYINOV*, Ali BAYRAKDAR
Istanbul Aydin University, Küçükçekmece, Istanbul, 34307, Turkey
ilhamhuseyinov@aydin.edu.tr (*Corresponding author*), bayrakdarali85@gmail.com

Abstract: Inventory optimization is a significant problem that is tied directly to financial gains. Its complexity has led to the development of new inventory models and optimization techniques. Evolutionary algorithms, particularly Pareto based evolutionary algorithms have been proven to be reliable for solving such problems. However, these evolutionary algorithms concentrate mostly on global search and have limited local search abilities. This leads to a poor convergence to the Pareto front. Among these algorithms the most studied are non-dominated sorting genetic algorithm-II and strength Pareto evolutionary algorithm2. This paper proposes a novel method that increases their convergence. The novelty is based on three techniques: Firstly, a time-based fitness assignment that favours solutions from previous generations is employed. Secondly, before the crossover process, the mating pool is updated with a positive bias towards better solutions. Finally, a more disruptive mutation scheme is used to prevent premature convergence. The novel algorithms were tested on a benchmark problem suite and two inventory problems. The performance of the algorithms is measured using hypervolume, generational distance and spacing metrics. The results illustrated by graphics indicate that the novel algorithms can obtain better convergence without increasing the time complexity.

Keywords: Inventory optimization, NSGA-II, SPEA2, Evolutionary algorithms.

1. Introduction

The inventory optimization process is a sophisticated problem that is characterized by many attributes. Accordingly, an inventory optimization problem can be solved using an inventory model with multiple parameters. An inventory model can have many objective functions, decision variables and constraints. Some examples of objective functions are optimizing profit, service rate, costs and warehouse occupancy. Meanwhile, warehouse capacity, ordering budget and shortage costs are examples of constraints. Order amount and selling price are among the most popular decision variables. The inventory models can also have parameters like procurement costs, demand type, number of products and product shelf life. Moreover, the inventory problem may interact with other areas of operational research. Investing in raw materials, production rate, service and maintenance activities, warehouse specifications, transportation, and choosing the supplier is among these areas. This makes inventory optimization an even more significant problem (Silver, 2008). This raises a need for developing general algorithms that can be applied to as many inventory models as possible. These algorithms are called metaheuristic algorithms. Metaheuristics known as Pareto-based evolutionary algorithms have garnered attention from the research community in particular. In some of these studies, the Pareto-based algorithms were enhanced with an addition of local search operators (Azuma et al., 2011) or reference systems (Khishtandar & Zandieh, 2017; Sadeghi et al., 2014). In some of these studies, the original algorithms were used (Chołodowicz & Orłowski, 2017; Huseyinov & Bayrakdar, 2019) and in some of them, the original algorithms were compared with swarm-based algorithms (Sanchez et al., 2010). In the studies in which the original algorithms were used (Chołodowicz & Orłowski, 2017; Huseyinov & Bayrakdar, 2019) and the studies in which the original algorithms were compared to swarm-based algorithms (Sanchez et al., 2010), they have been proved to be effective in solving the inventory problem. However, it was observed that their convergence is weak compared to their diversity (Coello Coello et al., 2002; Emmerich & Deutz, 2018).

This paper aims to improve the convergence of the most popular Pareto-based evolutionary algorithms – the non-dominated search genetic algorithm-II (NSGA-II) and the strength Pareto evolutionary algorithm2 (SPEA2), by introducing novel operators (Deb et al., 2000; Zitzler et al., 2001). The novelty is threefold: Firstly, a time-based fitness assignment that favours solutions from previous generations is employed. Secondly, before the crossover process, the mating pool is updated with a positive bias towards better solutions. Finally, a more disruptive mutation scheme is used to prevent premature convergence. The novel algorithms are tested on a benchmark problem and on two inventory models that have a wide area of application in real life as presented in multiple

https://doi.org/10.24846/v31i3y202203

studies. Similar models with products with finite shelf lives (Sharma, 2004; Zhang, 2010), objectives concerning general inventory management costs (Hnaien et al., 2016; Zhang, 2010) and general multi-objective models (Huseyinov & Bayrakdar, 2019; Khishtandar & Zandieh, 2017) are used in these studies. The performance of the algorithms is measured using hypervolume, generational distance and spacing metrics. The results illustrated by graphics indicate that the novel algorithms can obtain better convergence without increasing the time complexity.

The rest of the paper is organized as follows. Section 2 offers a general survey of the previous studies that addressed the problem of these algorithms. Section 3 presents preliminary concepts and the proposed algorithms. Definitions concerning Pareto optimization are given here. Also, the pseudocode and the explanation of the novel algorithms are presented. The three novelties proposed and their particular integration into the original algorithms are explained in detail. Section 4 describes the experiments through which the proposed algorithms are implemented. First, the problem-independent parameters of the novel algorithms are fine-tuned. Then the algorithms are applied to a test suite and two inventory problems. The results of the conducted experiments are discussed in Section 5. Here, it is proven that the novel algorithms improve convergence. Finally, in Section 6, the conclusion is presented.

2. Literature Review

Reviews of studies about the application of metaheuristic algorithms to different inventory models are as follows: The dynamical programming approach is applied by Minner (1997). In the study (Andersson & Melchiors, 2001), a heuristic algorithm was proposed to solve an inventory problem. However, the obtained solution is 40% better than the optimal solution. Sharma (2004) applied an analytical approach. The work of Chan et al. (2005) applied a differential evolution algorithm. Although the differential evolution algorithm performs better on single-objective problems, it still performed well in this study. The differential evolution algorithm was proven to be better than the MATLAB optimizer.

The study (Panda et al., 2008) presented a reduced gradient method for an inventory problem with imperfect products. However, it was not compared with any other algorithms. In the study of Huang & Lin (2010), a modified ant colony optimization algorithm was proposed. It outperformed the original ant colony algorithm and obtained a smaller total cost. The research (Shiguemoto & Armentano, 2010) presented a variant of tabu search with an enhanced ability for local search. The algorithm obtained optimal solutions within a short time frame.

Sanchez et al. (2010) compared the performance of SPEA2 and its modified version. Also, the well-known NSGA-II and particle swarm optimization (PSO) algorithms were used for an additional comparison. The modified SPEA2 obtained better quality solutions than the other algorithms. Zhang (2010) applied Lagrange multipliers to find Lagrange solutions. Then the author suggested a heuristic algorithm to construct solutions based on Lagrange solutions.

The study conducted by Azuma et al. (2011) applied a SPEA2 algorithm with local search and various standard genetic operations to an inventory problem with deterministic demands. The algorithm performed well but its performance for solving a problem with stochastic demand was not explored. Mousavi & Pasandideh (2011) employed the genetic algorithm to solve five numerical examples. The genetic algorithm proved to be accurate in solving these problems. Zhao et al. (2012) presented a local search algorithm that obtained optimally and near-optimal solutions. They used a technique that calculates an optimal solution from a group of neighbouring solutions. As the number of neighbouring solutions increases, the quality of the optimal solutions increases as well. However, this led to higher time complexity.

Sadeghi et al. (2014) carried out a performance comparison of four algorithms: NSGA-II, a hybrid NSGA-II with local search, a non-dominated ranking genetic algorithm (NRGA) and a hybrid NRGA with local search. The hybrid NSGA-II performed better than the other algorithms. However, the authors indicate a need for parameter tuning.

In the study (Hnaien et al., 2016), a branch-and-bound algorithm performed better than a heuristic algorithm. In the study (Lagos et al., 2016), an inventory model with objectives of minimizing warehouse location and general inventory management costs was solved by Pareto local search. It was concluded that Pareto local search is effective for such a problem. Khishtandar & Zandieh (2017), carried out a comparison between constrained reference-based non-dominated sorting genetic algorithm-II (C-R-NSGA-II) and constrained non-dominated sorting genetic algorithm-II (C-NSGA-II). C-R-NSGA-II proved to be better than C-NSGA-II. However, it was observed that the selection of reference points has a big impact on the quality of solutions.

Chołodowicz & Orłowski (2017) compared NSGA-II and SPEA2 for solving a bi-objective inventory problem. SPEA2 displayed a higher hypervolume than NSGA-II. A modified version of PSO and a simulated annealing algorithm (SAA) were compared by Pasandideh et al. (2017). The used PSO is a parameter-tuned variation of the original PSO. The modified PSO performed better than SAA.

In the study of Huseyinov & Bayrakdar (2019), a multi-objective inventory problem was solved with non-dominated sorting genetic algorithm-III (NSGA-III) and SPEA2. SPEA2 was found to perform better than NSGA-III.

Simić et al. (2019) divided the inventory problem into two parts. The first part of the problem was solved using the PSO algorithm and the second part was solved using pure adaptive search. The authors concluded that the optimization algorithms by themselves are insufficient, hence, there is a need for a decision-maker.

3. Preliminaries

3.1 Basic Concepts

A constrained multi-objective problem is concerned with minimizing objective functions by finding the right decision variable values under constraints. It is defined as minimizing $F(x)=(f_1(x), f_2(x), ..., f_m(x))$ while being subject to $h_i(x)=0$, $i=\{1,...,n\}$ and $g_j(x) \leq 0, j=\{1,...,z\}$, where $x=(x_1,...,x_k)^T$ is a k-dimensional vector from decision variable space Ω. The number of objectives is m.

For $x, y \in \Omega; x$ Pareto dominates y, if and only if $\forall n \in \{1,...,m\}, f_n(x) \leq f_n(y)$ and $\exists n \in \{1,...,m\}, f_n(x) < f_n(y)$. Pareto dominance is symbolized by $\prec$. Therefore $x \prec y$. For $x^* \in \Omega$, if and only if $\nexists x \in \Omega, x \prec x^*, x^*$ is a Pareto optimal solution. All Pareto optimal solutions are the Pareto optimal set (POS). The objective vectors yielded by mapping every member of the Pareto optimal set to the objective space through the objective functions are called the Pareto front (PF). The Pareto front is defined as $PF = \{F(x) \mid x \in POS\}$.

3.2 Evolutionary Algorithms

Currently, evolutionary algorithms are among the most popular methods used for solving inventory problems because they have some advantages over other metaheuristic algorithms (Emmerich & Deutz, 2018).

These advantages are:

- They are population-based;
- They find solutions in a single run;
- They allow greater parameter optimization.

Particularly, Pareto-based evolutionary algorithms like NSGA-II and SPEA2 are popular (Deb et al., 2000; Zitzler et al., 2001). Several studies (Khishtandar & Zandieh, 2017; Sadeghi et al., 2014; Sanchez et al., 2010) presented evidence in favour of evolutionary algorithms over other types of metaheuristic algorithms. In general NSGA-II and SPEA2 are performant when there are two objectives and few parameters. Also, they have diversity-preserving mechanisms that generate diverse Pareto fronts. However, they have poor convergence due to poor local search abilities. On the other hand, local search algorithms are prone to falling into local optima (Emmerich & Deutz, 2018).

Pareto-based evolutionary algorithms have two ranking mechanisms. Firstly, the solutions are ranked following their Pareto dominance. Dominant solutions are deemed better. Secondly, the solutions are ranked following their relative positions to other solutions. Solutions that map to the less crowded parts of the solution space are considered to be better alternatives. This is known as diversity preservation. Diversity preservation is used when comparing two solutions that have the same Pareto dominance level (Emmerich & Deutz, 2018). NSGA-II and SPEA2 are the most popular evolutionary algorithms which use this two-level ranking mechanism. They are dated and have been subject to many studies. However, they still stand as the golden standard for bi-objective evolutionary optimization.

3.3 Proposed Algorithms

In the proposed algorithms, in addition to the dominance rank and diversity preservation measures, a generation count is implemented. Every solution has a generation count which increases as the solution survives more generations. The generation count is used during binary tournament selection. If the two solutions being compared have the same dominance rank, the one with the higher generation count is selected. If their generation rank is also equal, then the solution mapping to the less crowded region of the objective space is selected. Newly discovered solutions have a generation count equal to zero. After a solution undergoes crossover the generation count is reset. In Figure 1, two objective functions are to be minimized.

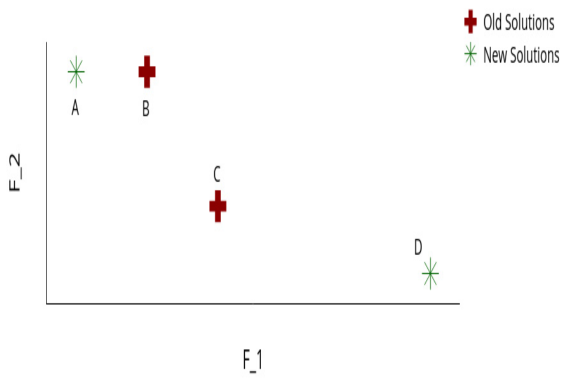

Figure 1. Comparison of old and new solutions

The new solutions in the figure are assumed to have a generation count equal to zero, while the old solutions have a generation count greater than zero. During the binary selection phase, if solutions A and B are compared, solution A is selected because it dominates solution B. On the other hand solution, A and solution C are both non-dominated solutions. Since solution C has a greater generation count, it will be selected over solution A. Solution A and solution D are both non-dominated solutions and they have the same generation count values. However, solution D rests on a less crowded part of the objective space, causing it to be selected over solution A. Fundamentally, this operation is a constrained application of the hill-climbing algorithm. The hill climbing algorithm is a local search method that directs the search towards the position of solutions discovered in previous iterations. It is very effective for local search, hence, it is used in this study. However, it is prone to fall into local optima (Whitley, 1994). Therefore, only a limited implementation is used: the generation counts come after the dominance rank in importance and it is reset after the associated solution undergoes crossover.

After the mating pool is filled, a tunable amount of fitter solutions is duplicated. Then, these duplicated solutions replace the same amount of random solutions in the mating pool. The random replacement enables the duplicated solutions to be replaced too. Therefore, the process is more likely to preserve diversity than just replacing the worst solutions with the best ones. A similar method is successfully used in the original (μ+λ)-ES algorithm. In the (μ+λ)-ES algorithm, after the offspring is created, the best solutions among both the parent and offspring population form the new parent population (Bäck et al. 1991). Implementing it directly in a multi-objective evolutionary algorithm would be detrimental because it can decrease diversity to the point of premature convergence. Therefore, only a tunable amount is replaced in the novel algorithms.

Following the mating process, the solutions which underwent crossover have their generation rank reset to zero. Then, after the standard mutation operator is completed, a polynomial mutation operator is applied to the same set of solutions. This is a different implementation of the Cross-generational, elitist selection, cataclysmic mutation (CHC) algorithm's mutation scheme. In the CHC algorithm, the solutions undergo heavy mutation to prevent premature convergence (Eshelman, 1991). In this study, two different mutation operators are used to further increase the effect of the mutation. Since the experiment is done with real-coded variables, the polynomial mutation is implemented along with the standard mutation operator.

3.3.1 Novel NSGA-II

In the proposed method, solutions yielded iteratively by the novel NSGA-II have a generation count value along with crowding distance and non-dominance rank values as explained earlier. Every solution has a generation count equal to zero when it is discovered by the algorithm. Then, during the fitness assignment of the solutions, their generation counter is increased by one. Therefore, the more generations the solution survives, the higher its generation count will be. When the solution undergoes crossover, the generation count is reset to zero. During the binary tournament selection, a new crowded comparison operator is used Firstly, the new crowded operator compares the non-domination ranks. In the case of a tie, the novel operator compares the generation counts. Then, the solution with the higher generation count is selected for the mating pool. If the generation counts of the compared individuals are the same, the novel operator compares the crowding distances and performs selection accordingly. After the selection process, the $r \times P$ number of the best individuals in the mating pool is duplicated and it replaces the $r \times P$ number of random individuals in the mating pool. P is the size of the mating pool and r is an adjustable parameter $r \in [0,1]$. The result $r \times P$ is rounded to the nearest whole number. When a solution undergoes crossover, both its generation count and one of its children become equal to zero. Finally, after the solutions undergo mutation, a polynomial mutation operator is invoked on the obtained pool of solutions. It is also worth noting that, while building the new population from the combination of parent and offspring population, the standard crowded comparison operator is used. Therefore, the sorting process has the same time complexity. The time complexity of assigning the generation count is $O(N)$ with N population size. The duplication process has time complexity $O(P)$. The time complexity of the algorithm is dominated by the non-dominating sorting process. Therefore, the novel algorithm has time complexity $O(MN^2)$. The pseudocode for the novel algorithm is presented in Algorithm 1.

Algorithm 1: Novel NSGA-II

Procedure: Novel NSGA-II
Input: Y
  //Y is the generation number according to which
  //the algorithm is going to be terminated
Initialize Z_0
  //Z_0 is the population
Set $W_0 = \emptyset$, $e = 0$
  //W_0 is the offspring pool
Calculate the fitness of Z_0
While ($e < Y$) **do**
  Perform selection,
  Update the mating pool concerning the r parameter;
  Perform crossover on Z_e and reset the generation count;
  Perform standard and polynomial mutation on Z_e to generate W_e;
  Perform sorting and fitness assignment with generation count on $Z_e \cup W_e$;
  Generate Z_{e+1} by using the crowded comparison operator;
  Set $e = e+1$;
End While
Return *Ze*

3.3.2 Novel SPEA2

The changes introduced in the novel SPEA2 are similar to the ones introduced in the novel NSGA-II. The individuals discovered by the novel SPEA2 have a generation count. The solutions have a generation count equal to zero when they are first yielded. During the fitness assignment phase of the population and the archive, the generation count of every solution is increased by one. Then the archive is filled using the standard methods used in the original SPEA2. During the binary tournament selection phase, in the case of two competing solutions having the same fitness value, the solution with the higher generation count value between the two is selected. If the fitness and generation values of these two solutions are the same, the one with the best density value is selected. Afterwards, the best $r \times P$ individuals in the mating pool are duplicated and they replace random $r \times P$ individuals in the mating pool. P is the size of the mating pool and r is an adjustable parameter. The product $r \times P$ is rounded to the nearest whole number and $r \in [0,1]$. When the solutions undergo crossover, they and their offspring will have a generation count equal to zero. Then, the mutation operator is applied in a standard way. Following this mutation process, a polynomial mutation operator is applied to the new pool of solutions. The assigning of the generation count has a time complexity of $O(N)$ where N is the population size. The duplication process has time complexity $O(P)$. Therefore, the time complexity of the new algorithm is the time complexity of density estimation: $O\left(N^2 logN\right)$. The pseudocode for the novel algorithm is presented in Algorithm 2.

Algorithm 2: Novel SPEA2

Procedure: Novel SPEA2
Input: *Y*
//*Y* is the generation number according to which the algorithm is going to be terminated
Initialize Z_0
//Z_0 is the population
Set $W_0 = \emptyset, T = \emptyset$, $u = 0$, $nds = 0$
//W_0 is the archive, *T* is the archive size
While ($u < Y$) **do**
Calculate the fitness of *Zu* and *Wu* with generation count;
nds = non-dominated solutions from *Zu* ∪ *Wu*;
If (*nds*) > *T* **Then**
Use truncation operator;
Else If (*nds*) < *T;*
Fill the remaining spots in *Wu* with the best-dominated solutions;
Else
Fill *Wu* with non-dominated points;
End If
Create the mating pool and update it concerning parameter *r*.
Perform recombination and reset the generation count of new solutions.
Perform mutation followed by polynomial mutation
Set $u = u+1$
End While
Return W_y

4. Experiment

4.1 Experimental Settings and Parameters

The experiment was conducted on a laptop computer with an Intel i7-7500u processor, 16 GB RAM and Windows 10 operating system. MOEA framework was used for the experiment (GitHub, 2020a). The coding was done with Java 9 programming language in Eclipse IDE. The graphs were plotted with the Plotly Python library (GitHub, 2020b). The parameters for the algorithms are given in Table 1.

Table 1. Parameter values

Parameter	NSGA-II	SPEA2	Novel NSGA-II	Novel SPEA2
p_c	0.9	0.9	0.9	0.9
p_m	0.2	0.2	0.2	0.2
T	250	250	250	250
Z	-	125	-	125

Crossover probability is represented by p_c. Mutation probability is represented by p_m. T is the population size and Z is the archive size.

4.2 Performance Metrics

The performance of an evolutionary algorithm is evaluated with performance metrics which are used to measure the convergence and diversity of the Pareto front. A metric can be used to measure convergence, diversity or both. Convergence metrics are used for measuring how close the solutions are to the Pareto front. On the other hand, diversity metrics are used for measuring how well the solutions are spread across the solution space. There is no consensus about which metric is the best. However, for a better evaluation, it is recommended to measure both the diversity and convergence (Laumanns et al., 2002). It is also worth noting that achieving good diversity and good convergence are usually conflicting goals. In this study, hypervolume, generational distance and spacing metrics are used.

Hypervolume measures both the convergence and diversity of the obtained solutions. It calculates the volume in the solution space which is covered by the solution set *N*. The cumulative volume of hypercubes constructed with each solution and a reference point gives the hypervolume value. A larger hypervolume value is then sought (Zitzler & Thiele, 1998). Generational distance is a convergence metric. It measures the average distance of the solution set from the Pareto front or another set of reference points. A smaller generational distance indicates better convergence (Coello Coello et al., 2002). Spacing is a diversity metric that measures how similar the relative distances between the non-dominated solutions are. A smaller spacing value indicates better diversity (Schott, 1995).

4.3 Validation

Two benchmark problems, Schaffer and Schaffer 2 have been solved with the two novel algorithms for finding the best value of the parameter r (Schaffer, 1985). The problems have been solved for r values within the range of 0.1 to 0.9 with a 0.1 step size. The algorithms have been run 10 times for each r value. The obtained average hypervolume values are presented in Figures 2 and 3.

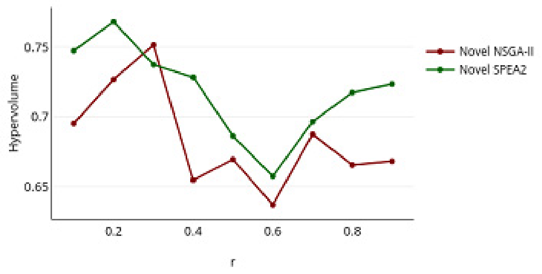

Figure 2. Hypervolume obtained for Schaffer

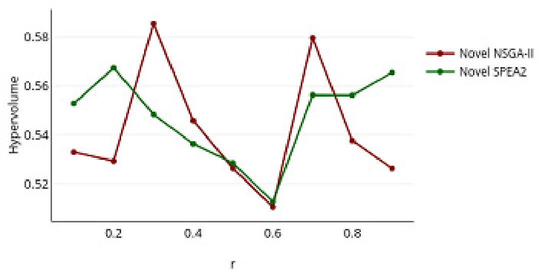

Figure 3. Hypervolume obtained for Schaffer2

The novel algorithms have the highest hypervolume value for the Schaffer problem in the 0.2-0.3 range. For Schaffer 2 problem, the novel algorithms have the lowest hypervolume value within the range 0.4 – 0.6. Novel NSGA-II has the highest hypervolume measurement for an r-value of 0.3. Novel SPEA2 has the highest hypervolume measurement for an r-value of 0.7. For these reasons, the r parameter is equal to 0.3 for the following experiments.

After setting the r-value, the novel algorithms have been compared with NSGA-II and SPEA2 for solving the well-known ZDT problems (Zitzler, Deb & Thiele, 2000). ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6 problems have been used for the experiment. ZDT5 uses binary-coded variables, hence, it is not used in studies conducted with real-coded variables.

The average, as well as the best generational distance and spacing values obtained, are given in Table 2 and Table 3, respectively. The results show that Novel NSGA-II has the best generational distance while Novel SPEA2 has the second best. However, NSGA-II has the best spacing value and SPEA2 has the second best.

Table 2. Spacing values

		Novel NSGA-II	Novel SPEA2	NSGA-II	SPEA2
ZDT1	Best	0.0613	0.0051	0.0475	0.0313
	Mean	0.1811	0.1476	0.1249	0.1153
ZDT2	Best	0.0037	0.0052	0.0029	0.0027
	Mean	0.1792	0.1705	0.1146	0.1375
ZDT3	Best	0.0045	0.0039	0.0032	0.0029
	Mean	0.1365	0.1278	0.0903	0.0925
ZDT4	Best	0.5341	0.0977	0.0035	0.0043
	Mean	2.9457	3.0192	2.5246	2.8486
ZDT6	Best	0.0309	0.0322	0.0215	0.0253
	Mean	0.2948	0.2805	0.2197	0.1651

Table 3. Generational distance values

		Novel NSGA-II	Novel SPEA2	NSGA-II	SPEA2
ZDT1	Best	0.2325	0.1859	0.2941	0.2475
	Mean	0.3773	0.3703	0.4219	0.3904
ZDT2	Best	0.5458	0.5722	0.5919	0.5936
	Mean	0.7282	0.7901	0.8464	0.8473
ZDT3	Best	0.1322	0.1408	0.1642	0.1571
	Mean	0.1953	0.1929	0.2188	0.2025
ZDT4	Best	3.0863	3.5971	4.1273	3.8952
	Mean	5.0995	5.6174	7.6564	6.2106
ZDT6	Best	1.4334	1.5662	1.6426	1.6799
	Mean	1.8753	2.1317	2.2367	2.8275

Therefore, the novel algorithms have a better convergence than the one of the standard algorithms. For solving a real-world problem, a decision-maker chooses the most suitable solutions from the Pareto front obtained by the optimization algorithms. If the decision-maker needs solutions closer to the optimal solutions, rather than a more diverse set of choices, the novel algorithms are a better choice. Optimization algorithms perform differently for different problems but the improvements in the algorithms usually carry on to real-world problems. The ZDT suite contains Pareto fronts with various shapes so it is a good indicator of real-world performance. The effectiveness of the novel algorithms for solving real-world problems is proven in the following subsection.

4.4 Inventory Problems

The novel algorithms, along with the original algorithms, are applied to two different inventory models. In the first model, the objective is to maximize both the profit and the service rate. There is a constraint on warehouse capacity. The decision variable is the order amount. In the second model, the objectives are to maximize the profit while minimizing the holding cost. There is again a constraint on warehouse capacity. Demand has uniform distribution for the two models. The parameters used in both models are as follows:

For product number $i = 1,..,n$

P : Expected profit

R : Average service rate

W : Warehouse capacity

L : Expected income

H : Expected holding cost

S : Expected shortage cost

O : Expected order cost

h_i : Percentage of holding cost of the ith product compared to Z_i

s_i : Percentage of shortage cost of the ith product compared to Z_i

o_i : Percentage of order cost of the ith product compared to Z_i

N : Product number

Z_i : The selling price of the ith product

Y_i : Order amount of the ith product

D_i : Stochastic demand for the ith product

$F_i(x_{.i})$: Demand probability mass function

θ_i : Minimum expected demand for the ith product

φ_i : Maximum expected demand for the ith product

The income is calculated with equation (1). Holding cost, shortage cost and order cost are calculated with equations (2), (3) and (4) respectively. Then, the total income is calculated with equation (5). The service rate is calculated with equation (6).

$$L = \sum_{i=1}^{N} (min(D_i, Y_i) Z_i F_i(x_i)) \tag{1}$$

$$H = \sum_{i=1}^{N} (Y_i Z_i F_i(x_i) h_i) \tag{2}$$

$$S = \sum_{i=1}^{N} \begin{Bmatrix} (D_i - Y_i) Z_i(x_i) s_i, if (D_i - Y_i) > 0 \\ 0, if (D_i - Y_i) \leq 0 \end{Bmatrix} \tag{3}$$

$$O = \sum_{i=1}^{N} Y_i Z_i F_i(x_i) o_i \tag{4}$$

$$P = L - H - S - O \tag{5}$$

$$R = (\sum_{i=1}^{N} \frac{min(D_i, Y_i)}{D_i}) / N \tag{6}$$

The first inventory model is given in equation (7) and the second inventory model is given in equation (8).

$$\begin{aligned} &Maximize \begin{Bmatrix} P \\ R \end{Bmatrix} \\ &Subject\ to \left(\sum_{i=1}^{N} Y_i \right) \leq W \end{aligned} \tag{7}$$

$$\begin{aligned} &Maximize\ P \\ &Minimize\ H \\ &Subject\ to (\sum_{i=1}^{N} Y_i) \leq W \end{aligned} \tag{8}$$

The dataset for the first inventory model is given in Table 4 and the dataset for the second inventory model is given in Table 5. It is assumed that every item takes one unit of space in the warehouse and the warehouse capacity is given as the total units it can hold. Each model is solved with novel NSGA-II, novel SPEA2, NSGA-II and SPEA2 for ten runs. The average, as well as the best generational distance and spacing values, are recorded.

Table 4. Dataset for the first inventory model

Product Number	1	2	3	4	5
Θ	11	23	24	30	17
Φ	200	292	168	251	209
Z	15	16	20	18	12
H	0.5	0.5	0.5	0.5	0.5
S	0.2	0.3	0.3	0.2	0.3
O	0.7	0.7	0.7	0.7	0.7
Warehouse Capacity:				900	

Table 5. Dataset for the second inventory model

Product Number	1	2	3	4	5
Θ	16	19	23	28	19
Φ	210	254	279	256	210
Z	15	16	20	18	12
H	0.6	0.6	0.6	0.6	0.6
S	0.3	0.3	0.3	0.3	0.3
O	0.7	0.7	0.7	0.7	0.7
Warehouse Capacity:					900

5. Results and Discussions

The generational distance values and the spacing values for each algorithm are given in Tables 6 and 7 respectively. First, the ANOVA test is applied to these values. Then Tuckey's HSD is applied as a post-doc test. Results that are statistically better are bolded in the tables.

The average computational time for each algorithm is recorded. ANOVA and Tuckey's HSD tests are applied to the yielded values. The results reveal that there is not any statistically difference between the four algorithms. The computational times of the four algorithms are displayed in Figure 4.

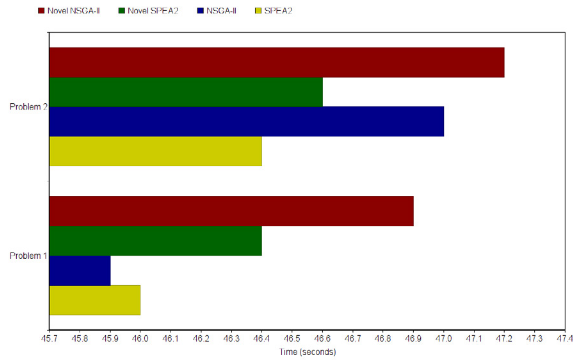

Figure 4. Computational times for solving the inventory models

The results reveal that the novel NSGA-II algorithm has a better convergence than one of the other three algorithms and the novel SPEA2 algorithm has the second-best convergence.

However, NSGA-II has a better diversity than one of the other three algorithms and SPEA2 has the second-best diversity. Between the novel NSGA-II algorithm and the novel SPEA2 algorithm, the novel SPEA2 algorithm has better diversity. The novel NSGA-II algorithm and the novel SPEA2 algorithm have a higher computational time than one of the original algorithms. However, the difference is statistically insignificant.

The solutions obtained by any of the original or novel algorithms are evaluated by a decision-maker, and the most suitable ones are chosen. The novel algorithms are better for use cases in

Table 6. Generational distance values

		Novel NSGA-II	Novel SPEA2	NSGA-II	SPEA2
First Inventory Problem	Best	**0.2371**	0.4212	0.9343	0.7453
	Mean	**0.3075**	0.6345	1.2123	0.8345
Second Inventory Problem	Best	**0.1523**	0.4219	0.6207	0.6123
	Mean	**0.2363**	0.6242	0.9935	0.8513

Table 7. Spacing values

		Novel NSGA-II	Novel SPEA2	NSGA-II	SPEA2
First Inventory Problem	Best	0.6429	0.5317	**0.1528**	0.3382
	Mean	0.7347	0.6966	**0.1773**	0.4314
Second Inventory Problem	Best	0.6199	0.5128	**0.2247**	0.3782
	Mean	0.6753	0.6766	**0.2864**	0.4978

which the decision-maker needs more precision by having solutions close to the true Pareto front. On the other hand, original algorithms are better for use cases in which the decision-maker needs a more diverse set of alternatives. Moreover, the novel algorithms are still evolutionary algorithms. Thus, they still retain all the advantages the evolutionary algorithms have over other algorithms while having improved convergence, which is a common weak spot of evolutionary algorithms.

6. Conclusion

Evolutionary algorithms are widely used to solve complex optimization problems such as the inventory problem. Pareto-based evolutionary algorithms are the most popular evolutionary algorithms. NSGA-II and SPEA2 are the most established and reliable algorithms among Pareto-based evolutionary algorithms. However, NSGA-II and SPEA2 have relatively poor convergence like most Pareto-based algorithms. To improve the convergence of these two algorithms, time-based fitness assignment and selection strategies are employed. Also, a novel mutation scheme is used to prevent premature convergence. The novel algorithms are validated with the well-known ZDT test problem suite. The results indicate that the novel algorithms have better convergence but worse diversity than the ones of the original algorithms. Then two inventory models with two objectives each are formulated. The novel algorithms are shown to have better convergence and worse diversity than the ones of the original algorithms for optimizing these two models. The models are bi-objective models and models with more than two objectives can be subject to future research. The original algorithms are mostly used in bi-objective optimization and they are known to perform poorly in problems with three or more objectives. Therefore, the novel algorithms can experiment on problems with three or more objectives to observe whether or not they have any improvements over the original algorithms.

In general, optimization algorithms display different performances for solving different optimization problems. This is mostly due to the shape of the true Pareto in front of the given problem. Therefore, there is a lot of room for future research on different inventory models. Additionally, different test problems can be used to test novel algorithms. Moreover, different performance metrics can be employed.

REFERENCES

Andersson, J. & Melchiors, P. (2001). A two-echelon inventory model with lost sales, *International Journal of Production Economics*, *69*(3), 307-315.

Azuma, R. M., Coelho, G. P. & Von Zuben, F. J. (2011). Evolutionary multi-objective optimization for the vendor-managed inventory routing problem. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, (pp. 1457-1464).

Bäck, T., Hoffmeister, F. & Schwefel, H. P. (1991). A survey of evolution strategies. In *Proceedings of the Fourth International Conference on Genetic Algorithms* (pp. 2-9).

Chan, F., Routroy, S. & Kodali, R. (2005). Differential evolution algorithm for supply chain inventory planning, *Journal of Manufacturing Technology Management*, *16*(1), 7-17.

Chołodowicz, E. & Orłowski, P. (2017). Comparison of SPEA2 and NSGA-II Applied to Automatic Inventory Control System Using Hypervolume Indicator, *Studies in Informatics and Control*, *26*(1), 67-74. DOI: 10.24846/v26i1y201708

Coello Coello, C. A., Van Veldhuizen, D. A. & Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Optimization Problems*. Springer, New York.

Deb, K., Agrawal, S., Pratap, A. & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International Conference on Parallel Problem Solving from Nature* (pp. 849-858).

Emmerich, M. T. & Deutz, A. H. (2018). A tutorial on multiobjective optimization: fundamentals and evolutionary methods, *Natural Computing*, *17*(3), 585-609.

Eshelman, L. J. (1991). The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In Rawlins, G. (Ed.), *Foundations of Genetic Algorithms*, *vol. 1*, 265-283.

GitHub (2020a). *MOEAFramework*. Available at: <https://github.com/MOEAFramework/MOEAFramework>, last accessed: 5th of January, 2020.

GitHub (2020b). *Ploty*. Available at: <https://github.com/plotly>, last accessed: 12th of January, 2020.

Hnaien, F., Dolgui, A. & Wu, D. D. (2016). A single-period inventory model for a one-level assembly system with stochastic lead times and demand, *International Journal of Production Research*, *54*(1), 186-203.

Huang, S. H. & Lin, P. C. (2010). 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.

Huseyinov, I. & Bayrakdar, A. (2019). Performance evaluation of NSGA-III and spea2 in solving a multi-objective single-period multi-item inventory problem. In *2019 4th International Conference on Computer Science and Engineering (UBMK)*, (pp. 531-535).

Khishtandar, S. & Zandieh, M. (2017). Comparisons of some improving strategies on NSGA-II for multi-objective inventory system, *Journal of Industrial and Production Engineering*, *34*(1), 61-69.

Lagos, C., Vega, J., Guerrero, G. & Rubio, J. M. (2016). Solving a Novel Multi-Objective Inventory Location Problem by means of a Local Search Algorithm, *Studies in Informatics and Control*, *25*(2), 189-194. DOI: 10.24846/v25i2y201606

Laumanns, M., Thiele, L., Deb, K. & Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization, *Evolutionary Computation*, *10*(3), 263-282.

Minner, S. (1997). Dynamic programming algorithms for multi-stage safety stock optimization, *Operations-Research Spektrum*, *19*(4), 261-271.

Mousavi, S. M. & Pasandideh, S. H. (2011). A multi-periodic multi-product inventory control problem with discount: GA optimization algorithm, *Journal of Optimization in Industrial Engineering, 4*, 37-44.

Panda, D., Kar, S., Maity, K. & Maiti, M. (2008). A single period inventory model with imperfect production and stochastic demand under chance and imprecise constraints, *European Journal of Operational Research*, *188*(1), 121-139.

Pasandideh, S. H. R., Niaki, S. T. A. & Keshavarzi, A. A. (2017). A two-echelon single-period inventory control problem with market strategies and customer satisfaction, *Journal of Uncertain Systems*, *11*(1), 18-34.

Sadeghi, J., Sadeghi, S. & Niaki, S. T. A. (2014). A hybrid vendor managed inventory and redundancy allocation optimization problem in supply chain management: An NSGA-II with tuned parameters, *Computers & Operations Research*, *41*, 53-64.

Sanchez, D., Amodeo, L. & Prins, C. (2010). Meta-heuristic approaches for multi-objective simulation-based optimization in supply chain inventory management, *Artificial intelligence Techniques for Networked Manufacturing Enterprises Management*, 249-269. Springer.

Schaffer, J. D. (1985). Multiple objective optimizations with vector evaluated genetic algorithms. In *Proceedings of the First International Conference on Genetic Algorithm* (pp. 93-100).

Schott, J. R. (1995). *Fault-Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*. Thesis (M.S.) No. AFIT/CI/CIA-95-039. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.

Sharma, S. (2004). Optimal production policy with shelf life including shortages, *Journal of the Operational Research Society*, *55*(8), 902-909.

Shiguemoto, A. L. & Armentano, V. A. (2010). A tabu search procedure for coordinating production, inventory and distribution routing problems, *International Transactions in Operational Research*, *17*(2), 179-195.

Silver, E. A. (2008). Inventory management: An overview, Canadian publications, practical applications and suggestions for future research, *INFOR: Information Systems and Operational Research*, *46*(1), 15-27.

Simić, D., Svirčević, V., Ilin, V., Simić, S. D. & Simić, S. (2019). Particle swarm optimization and pure adaptive search in finish goods' inventory management, *Cybernetics and Systems*, *50*(1), 58-77.

Whitley, D. (1994). A genetic algorithm tutorial, *Statistics and Computing*, *4*(2), 65-85.

Zhang, G. (2010). The multi-product newsboy problem with supplier quantity discounts and a budget constraint, *European Journal of Operational Research*, *206*(2), 350-360.

Zhao, X., Qiu, M., Xie, J. & He, Q. (2012). Computing (r, Q) policy for an inventory system with a limited sharable resource, *Computers & Operations Research*, *39*(10), 2368-2379.

Zitzler, E., Deb, K. & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation*, *8*(2), 173-195.

Zitzler, E., Laumanns, M. & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm, *TIK-report 103*, pp. 19.

Zitzler, E. & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms – A comparative case study. In *International Conference on Parallel Problem Solving from Nature* (pp. 292-301).