

Application of POMAC to the Multiobjective 0/1 Knapsack Problem

Rié Komuro

E. David Ford



NRCSE

Technical Report Series

NRCSE-TRS No. 062

May 1, 2001

The NRCSE was established in 1997 through a cooperative agreement with the United States Environmental Protection Agency which provides the Center's primary funding.

Application of POMAC to the Multiobjective 0/1 Knapsack Problem

Rié Komuro
Dept of Applied Mathematics
Box 352420
University of Washington
Seattle, WA 98195-2420

E. David Ford
College of Forest Resources
Box 352100
University of Washington
Seattle, WA 98195-2100

1 Introduction to the Multiobjective 0/1 Knapsack Problem

1.1 Definition of the Multiobjective 0/1 Knapsack Problem

The knapsack problem is hard to solve although the problem description is simple. The test is really of the evolutionary computation software, because the test function is understandable and easy to formulate, and the experiments are repeatable and verifiable.

In this work, we use the knapsack problem as a test of the Pareto Optimal Model Assessment Cycle evolutionary computation software (Ford [2], Reynolds [3]), and we compare this software with other published software.

Generally, a 0/1 knapsack problem consists of a set of items, weight and profit associated with each item, and a maximum capacity of the knapsack. The goal is to find a set of items maximizing the total profits, as well as satisfying the weight constraints, i.e., the total weight for each knapsack does not exceed the given capacity.

Definition of the multiobjective 0/1 knapsack problem considered here is as follows:

Given a set of items x_j ($j = 1, 2, \dots, m$), each with weight $w_{1,j}, w_{2,j}, \dots, w_{n,j}$, and profit $p_{1,j}, p_{2,j}, \dots, p_{n,j}$ associated that can be assigned to knapsack 1, knapsack 2, \dots , knapsack n , and an upper bound c_1, c_2, \dots, c_n for the weight capacity of the knapsacks, find a subset of items which maximizes the total of the profits in the subset, yet all selected items fit into the knapsacks, i.e.,

find a vector $x = (x_1, x_2, \dots, x_m) \in \{0, 1\}^m$ such that

$$x_j = \begin{cases} 1 & \text{if object } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

maximizing $f(x) = (f_1(x), f_2(x), \dots, f_n(x))$ with

$$f_i(x) = \sum_{j=1}^m (p_{i,j} \cdot x_j),$$

subject to

$$\forall i \in \{1, 2, \dots, n\}, \quad \sum_{j=1}^m (w_{i,j} \cdot x_j) \leq c_i,$$

where

$$\begin{aligned} p_{i,j} &= \text{profit of item } j \text{ according to knapsack } i \\ w_{i,j} &= \text{weight of item } j \text{ according to knapsack } i \\ c_i &= \text{capacity of knapsack } i. \end{aligned}$$

Note: The values of weight and profit depend on not only the item, but also the knapsack it might assigned to.

1.2 Problem Setting

Comparisons of evolutionary computation software in providing solutions to the knapsack problem are published on the web. We chose to examine the case of 2 objectives (knapsacks) with 100 items, and to compare the results to ones by other methods, which are introduced later. The test data $p_{i,j}$, $w_{i,j}$ and c_i as used for other methods was obtained. Uncorrelated profits and weights, $p_{i,j}$ and $w_{i,j}$, are random integers in the interval $[10, 100]$, and they are defined on

www.tik.ee.ethz.ch/~zitzler/testdata.html.

Half of the items were anticipated to be in the optimal solution, that is,

$$c_i = 0.5 \sum_{j=1}^m w_{i,j}.$$

2 Introduction to POMAC

2.1 Algorithm of POMAC

The brief flow of the POMAC algorithm (Ford [2]) is as follows:

Let g , P_g and H_g be a current generation, a population of individuals for g and a Historical Pareto Frontier¹ for g , respectively.

Step 0: Set $g = 0$, and generate an set of initial populations P_0 .

Step 1: Assess the criteria of each simulation in P_g , and referring to the nondominated rank, assign a niche count² and a fitness value³ for each simulation.

Step 2: With these current generation results, update the Pareto Historical Frontier H_{g-1} to H_g .

Step 3: Based on fitness values, select parents for breeding the next generation.

Step 4: Apply crossover⁴ and mutation⁵ to the selected parents in *Step 3*.

Step 5: If g reaches the maximum number of generation, or all criteria are satisfied, then stop; else $g := g + 1$ and go to *Step 1*.

¹A Historical Pareto Frontier is the Pareto optimal set of the simulation results generated so far.

²A niche count is the number of neighbors an individual has in the current population, where neighbors are defined by their having similar (but not

necessarily identical) assessment vectors.

³A fitness value is calculated based on its nondominated ranking; this is then adjusted for the individual's niche count.

⁴A crossover exchanges some of the parameter values within the parameterization vector of one parent with some of the parameter values of another parent, producing two new individuals.

⁵A mutation incrementally changes, by a randomly chosen amount, a randomly selected parameter value from the parent's parameterization. The mutation for knapsack problem is, however, a slight different from this general case since each parameter takes values on only 0 or 1. We explain later how we set the mutation for the knapsack problem (Section 4.2 - 4.8).

2.2 Specialty of the Knapsack Problem

As originally written, POMAC could not be applied to the knapsack problem without modifications because of the following two reasons:

1. each parameter takes values of only 0 or 1;
2. each knapsack has an upper limit of the total weight defined above as c_i ($i = 1, 2, \dots, n$).

In consideration of these issues, for the initial population, only 0 or 1 was chosen randomly when P_0 was generated. Once P_0 was defined, each population member in P_0 was checked if it satisfied the weight constraints, i.e., $\sum_{j=1}^m (w_{i,j} \cdot x_j) \leq c_i$. If not, items were removed randomly from the knapsacks until all of the weight criteria were satisfied.

For the case after generation 0, we will explain later how to deal with these issues (Section 4.2 - 4.8).

The following diagram is an example of a possible status for the 100 item 2 knapsack problem:

		item 1	item 2	item 3	item 4		item 99	item 100
		$x_1 = 1$	$x_2 = 0$	$x_3 = 0$	$x_4 = 1$		$x_{99} = 1$	$x_{100} = 1$
knapsack1	weight	$w_{1,1}$	0	0	$w_{1,4}$		$w_{1,99}$	$w_{1,100}$
	profit	$p_{1,1}$	0	0	$p_{1,4}$...	$p_{1,99}$	$p_{1,100}$
knapsack2	weight	$w_{2,1}$	0	0	$w_{2,4}$		$w_{2,99}$	$w_{2,100}$
	profit	$p_{2,1}$	0	0	$p_{2,4}$		$p_{2,99}$	$p_{2,100}$

In this case, item 1, item 4, \dots , item 99, item 100 are selected, and item 2, item 3, \dots are not. Therefore, the total weights are

$$\begin{aligned} w_{1,1} + w_{1,4} + \dots + w_{1,99} + w_{1,100} &\leq c_1 && \text{for knapsack 1} \\ w_{2,1} + w_{2,4} + \dots + w_{2,99} + w_{2,100} &\leq c_2 && \text{for knapsack 2,} \end{aligned}$$

where c 's are maximal capacity for each knapsack, and the total profits are

$$\begin{aligned} p_{1,1} + p_{1,4} + \dots + p_{1,99} + p_{1,100} &&& \text{for knapsack 1} \\ p_{2,1} + p_{2,4} + \dots + p_{2,99} + p_{2,100} &&& \text{for knapsack 2.} \end{aligned}$$

3 Other Methods for Comparison

We tried seven different POMAC modified algorithms to apply to the 100 item 2 knapsack problem. For each trial, we plotted the nondominated total profits, (total profit in knapsack 1 vs. total profit in knapsack 2) along with the *frontiers* obtained by FFGA, SO-5, RAND, NSGA and SPEA, which are introduced next. Though larger population size was chosen for these methods, we tried size 50 for shorter running time. After we saw if the trial was successful, we executed with larger population size.

3.1 FFGA

FFGA, which was developed by Fonseca and Fleming (Fonseca [1]), is a rank-based fitness assignment methods for multiobjective genetic algorithms.

3.2 SO-5

SO-5 is a single-objective evolutionary algorithm (EA) that terminates after 500 generations in every single optimization run.

3.3 RAND

RAND randomly generates a certain number of individuals per generation, according to the rate of crossover and mutation (though neither crossover, mutation nor selection are performed).

3.4 NSGA

NSGA (Nondominated Sorting Genetic Algorithm, Srinivas [4]) is a multi-objective EA. As explained in Zitzler [5], for this method, the fitness assignment is carried out in several steps. In each, the nondominated solutions constituting a nondominated front are assigned the same dummy fitness value. These solutions are shared with their dummy fitness values (phenotypic sharing on the decision vectors) and ignored in the further classification process. Finally, the dummy fitness is set to a value less than the smallest shared fitness value in the current nondominated front. Then the next front is extracted. This procedure is repeated until all individuals in the population are classified. POMAC is based on this algorithm.

3.5 SPEA

SPEA (Strength Pareto Evolutionary Algorithm) is also a multiobjective EA, which was introduced by Zitzler [5] as follows:

On one hand, similarly to other multiobjective EAs, it

1. Stores the nondominated solutions found so far externally,
2. Uses the concept of Pareto dominance in order to assign scalar fitness values to individuals, and
3. Performs clustering to reduce the number of nondominated solutions sorted without destroying the characteristics of the trade-off front.

On the other hand, SPEA is unique in four respects:

1. It combines the above three techniques in a single algorithm.
2. The fitness of an individual is determined only from the solution stored in the external nondominated set; whether members of the population dominate each other is irrelevant.

3. All solutions in the external nondominated set participate in selection.
4. A new niching method is provided in order to preserve diversity in the population; this method is Pareto-based and does not require any distance parameter (like niche radius for sharing).

We chose the first three methods as the representative of different types of algorithms. The NSGA was chosen, because the POMAC was developed based on this method. The SPEA was introduced as the best EA to find the Pareto optimal solutions for the knapsack problems so far. The results for these five methods are published on the web (www.tik.ee.ethz.ch/~zitzler/testdata.html).

4 POMAC's Applications to the Knapsack Problem

4.1 Parameter Settings

To obtain the Pareto frontier of profits for the 100 item 2 knapsack problem by POMAC to be as good as those by other methods, we tried several different rates and combinations of crossover and mutation with 50 populations. We now, introduce seven trials where these were tested. For most of the trials, the parents receiving crossover had a proportion of 0.67 at the beginning, and it decreased to 0 linearly over the number of generations; otherwise, we state for each case. For most of the cases, we also applied mutation, and explain how for each time since it varies. A set of parents for breeding was selected depending on the fitness value, and which parents took crossover or mutation was determined at random.

4.2 First Trial

Considering the two issues mentioned in Section 2.2, 0/1 parameter values and the upper limits of the total weights, as the first step, we modified the original POMAC so that it could apply to the knapsack problem properly.

For crossover, using a random 0/1 generator, we determined whether each item was selected to be swapped or not. In other words, if a random 0/1

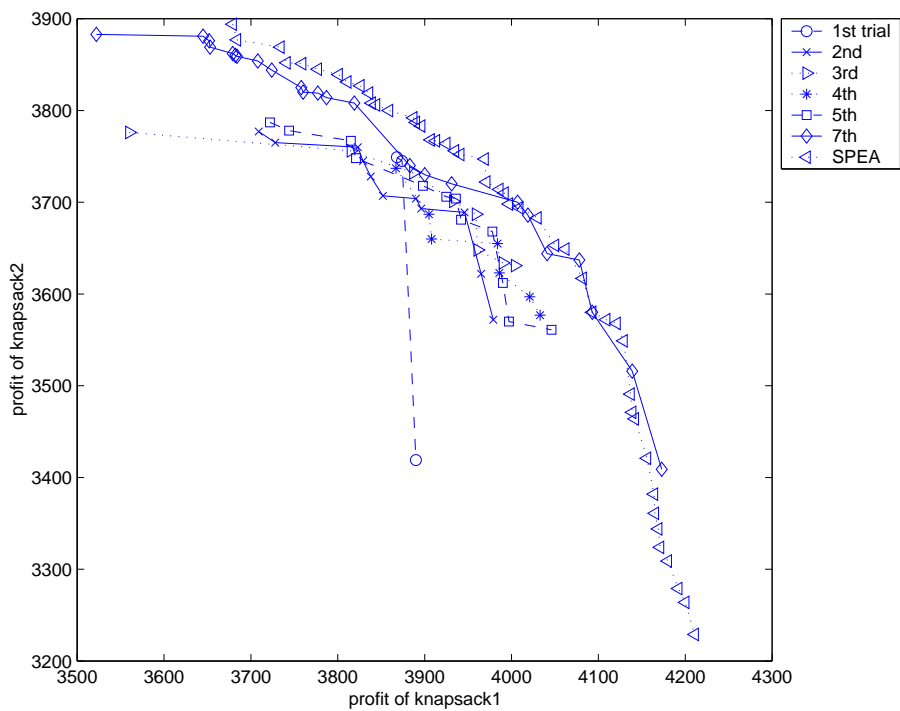


Figure 1: POMAC frontiers of the first through seventh trials at 500th generation. The population size is 50 except the seventh trial. For the seventh trial, it is 100.

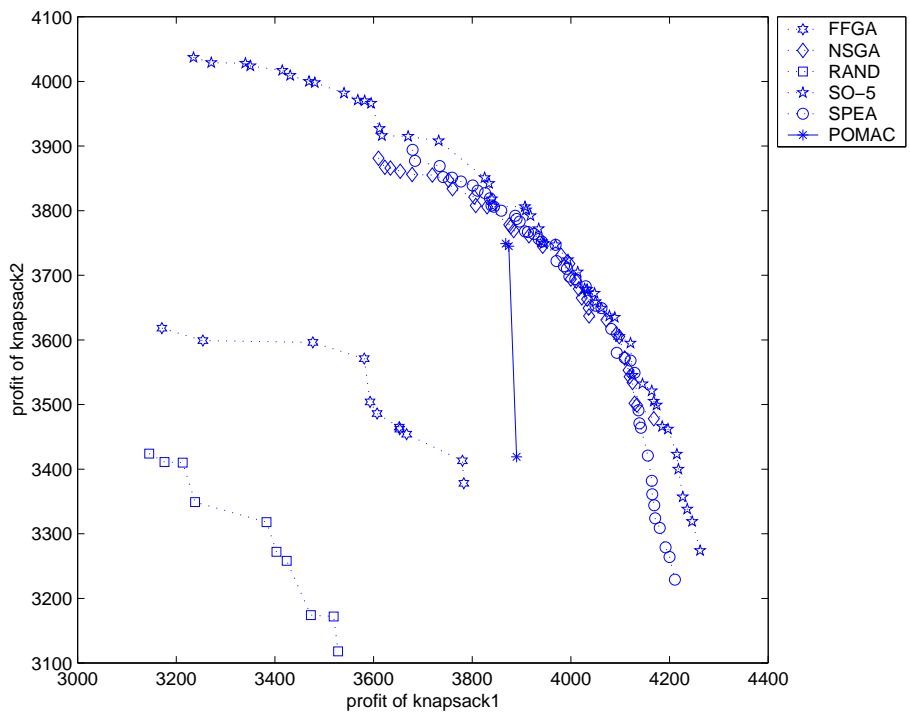


Figure 2: Result of POMAC's first trial (section 4.2) with other methods' Pareto frontiers at 500th generation. The Pareto frontier of POMAC consists of only a few individuals.

generator returned 1, then the item values x_j for *parent 1* and *parent 2* were exchanged, else they were not, where *parent 1* and *parent 2* were the currently examined paired parents. Swaps of anywhere from single items to 99 items were allowed.

After the crossover, we checked if the weight constraints were satisfied for each of the two children, which were newly generated from *parent 1* and *parent 2*. If not, we randomly chose an item and removed it from the knapsacks; for example, if one of the children, say *children 1*, broke the weight constraints, then the item of *children 1* which had value 1 was picked at random, and its value was changed to 0. This procedure was kept until the total weights in all knapsacks became less than or equal to their upper limits.

For mutation, only one item was chosen randomly, and its value was changed to 1 if it was 0, or to 0 otherwise. We set only one item to be changed, because mutation should bring a slight change of an individual to conserve inheritance. After this alternation, we checked over the weight constraints, and adjusted if necessary, which was exactly the same procedure done for crossover.

The result is shown on Fig. 2. The POMAC frontier (a set of the nondominated solutions) consists of only a few individuals, which means that this method gives only a few Pareto optimal solutions. Therefore, from the point of the Pareto frontier, this method was not successful although all individuals satisfied the weight constraints after *Step 5* of POMAC. This occurred probably because of the following reason. After crossover or mutation, in order to meet the weight constraints, we removed items for some of the children. This procedure worked for the weight capacities, but, on the other hand, made the total profit less and less. This could be the reason why this trial did not return a successful result.

4.3 Second Trial

The Pareto frontier obtained on the previous trial was considerably worse than that calculated by SPEA. This seemed to be affected by the procedure making each population meet the weight constraints after crossover or mutation. To fix this problem, we reconsidered the adjustment for the weight constraints.

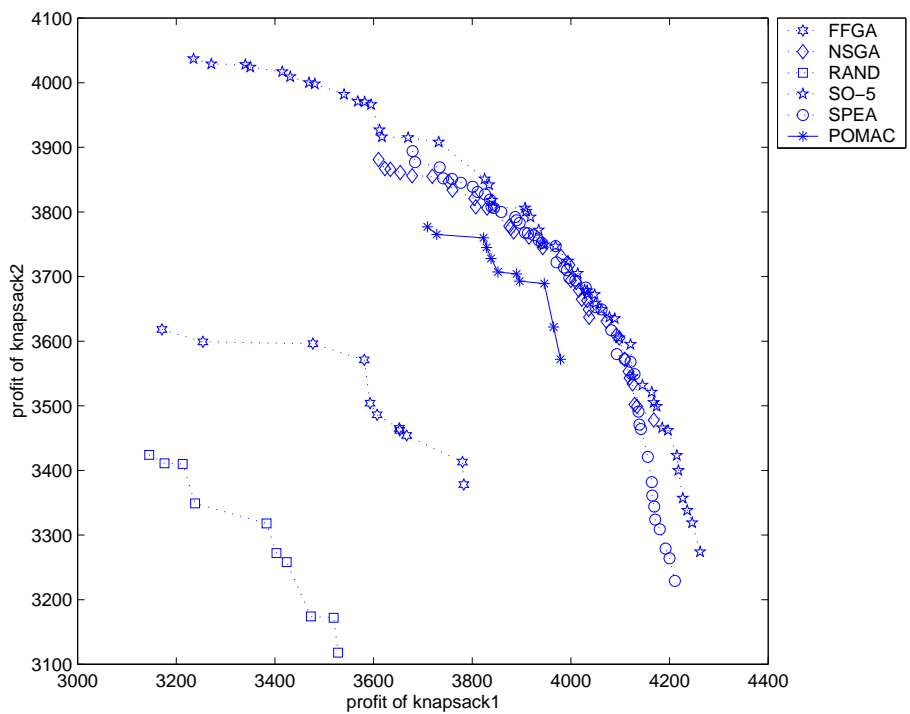


Figure 3: Result of POMAC's second trial (section 4.3) with Pareto frontiers of other method at 500th generation. Compared to the first trial, the frontier of POMAC was improved since it consists of more individuals.

The method of crossing over remained the same as for the first trial. After crossover, if a newly generated parent p did not satisfy the weight constraints, first, we randomly searched two different items x_r and x_a , where x_r was to be removed, and x_a was to be added, i.e., $x_r = 1$ and $x_a = 0$. If this exchange ($x_r = 0$ and $x_a = 1$) guaranteed that parent p met all of the weight constraints, the exchange was executed. If we could not find such as x_r and x_a , then similarly to the first trial, we removed items randomly until parent p met all the weight constraints. After that, to make the total profits larger, we randomly searched an item to be added such as didn't break the weight constraints. We kept the search to add an item as long as the weight constraints were satisfied.

For mutation, we first randomly picked an item that was not in the knapsacks, and checked if adding it to all of the knapsacks broke the weight constraints. If not, we added it, and the mutation was completed. Otherwise, we kept searching, up to 100 times, which was the number of items. One hundred searches could be sufficient since most of the items were checked by then.

Actually, we found that the possibility that an item could be added this way was low. On the failure of this procedure, we randomly picked two different items, one of which was to be removed, and another of which was to be added. If these removing and adding maintained the weight constraints, mutation was completed. Otherwise, we tried to find a pair of items up to 9900 times, which was the number of possible pairs with 100 items. 9900 searches could be sufficient since most of the pairs were checked by then. If we still could not find a pair, we removed items at random until all of the weight criteria were satisfied.

The result is shown on Fig. 3. Even though the POMAC frontier does not reach to those of SPEA and others, this trial can be said improved over the first trial since the POMAC frontier consists of more individuals. The trial for adding items after removing items, achieved a better result, compared to just removing items, which was carried out for the first trial. This trial, however, had a drawback on inheritance. Modifying individuals much after breeding could bring weaker inheritance.

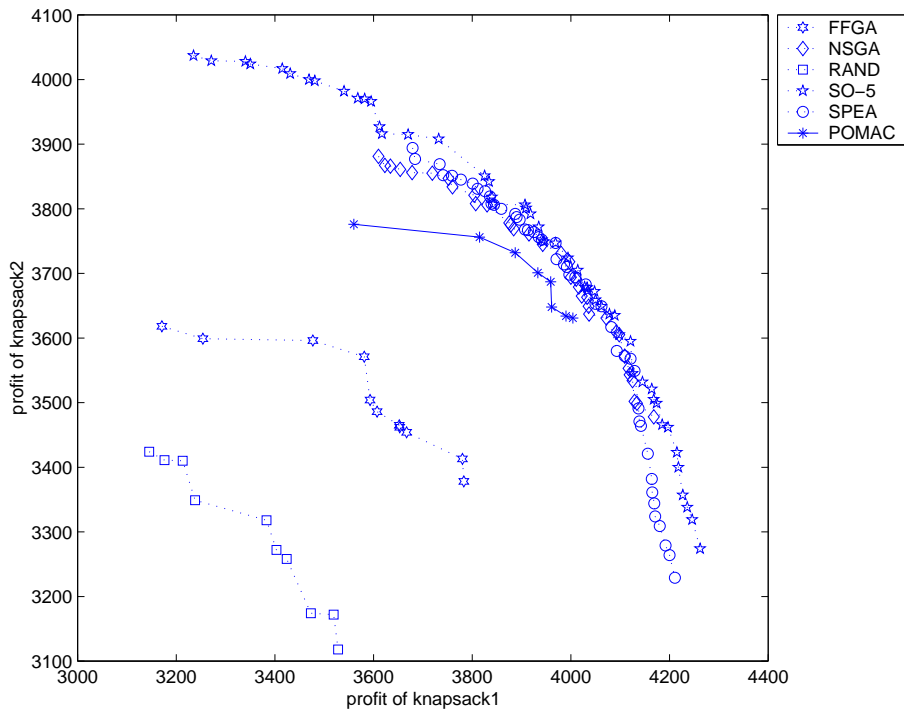


Figure 4: Result of POMAC's third trial (section 4.4) with Pareto frontiers of other methods at 500th generation. Compared to the previous trial, this result did not bring much improvement from the point of the number of frontier members as well as of the distance between the frontiers of POMAC and SPEA.

4.4 Third Trial

Next, to reduce influence by the drawback of the previous trial, we tried a different crossover. Mutation was same as the second trial.

For crossover, each item was determined whether swapped or not, in the same manner as the previous two methods. Whenever crossing over an item, a pair of parents was checked to see if it satisfied the weight constraints. If it did, the pair was stored and overwrote the previously stored pair. After the completion of crossing over items, i.e., all the 100 items were scanned, if the both of the children (the pair created from the parents) could fit into the knapsacks, then, of course, they became parents for the next generation. Otherwise, both of them were overwritten by the most recently stored parents, which met the weight constraints. Since the stored pair was created on the process for crossover, it did inherit the nature of the parents.

Actually, this trial had a problem on randomness. We always started from item 1 ascending to 100 to make a determination for swapping. Therefore, the children were usually the pair created after the item having larger index, such as 90, was checked for exchange.

The result is shown on Fig. 4. Compared to the previous trial, this did not bring much improvement from the point of the number of frontier members as well as of the distance between POMAC and SPEA frontier. We inferred from the result that changing adjustment for the weight constraints did not solve the problem, and that the real problem lied in the method of crossover or mutation.

4.5 Fourth Trial

Up till now, whether crossover was to take place was determined randomly for each item. So, ninety-nine items could be swapped between a pair of parents. Since each item value took only 0 or 1, the more swaps could generate very different children from their parents. Such children did not receive much inheritance from the parents. We considered crossover to generate more inherited children this time.

For this trial, we randomly picked a pair of items (j_1, j_2) available to swap

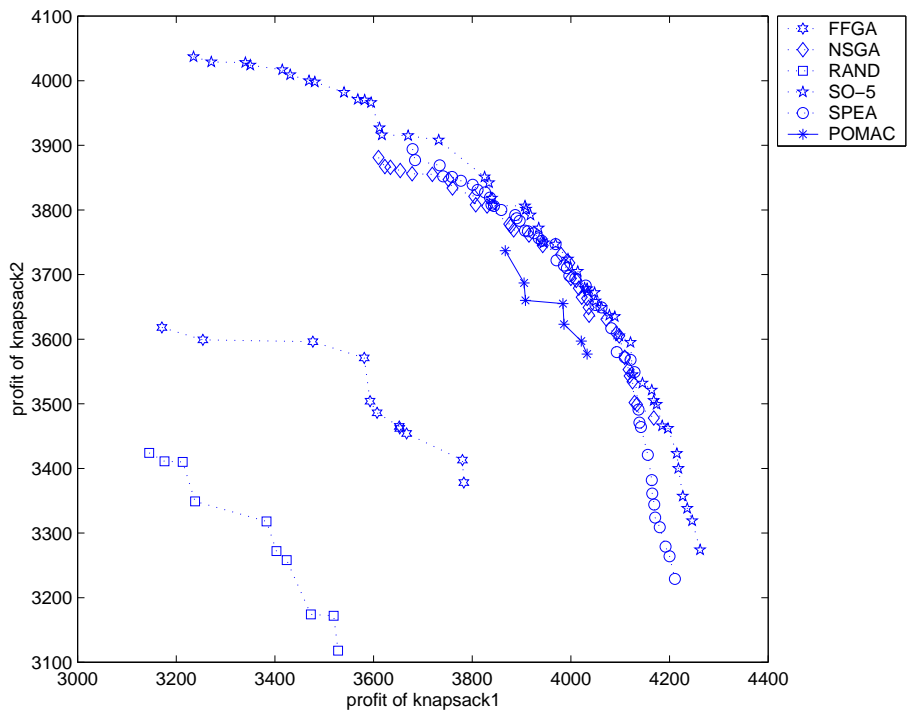


Figure 5: Result of POMAC's fourth trial (section 4.5) with the Pareto frontiers of other methods at 500th generation. There seems some bias toward knapsack 1 although the code was written without any bias.

the values between two parents, i.e., the case:

	item j_1	item j_2
parent 1 :	$\dots, 0, \dots,$	$1, \dots$
parent 2 :	$\dots, 1, \dots,$	$0, \dots$

or

	item j_1	item j_2
parent 1 :	$\dots, 1, \dots,$	$0, \dots$
parent 2 :	$\dots, 0, \dots,$	$1, \dots$

We executed the exchange if the weight constraints would be satisfied. Otherwise, we kept searching a pair of items (j_1, j_2) so that *parent 1* and *parent 2* met the weight constraints after the swap. After items j_1 and j_2 were swapped successfully, we added an available item chosen randomly if possible. We had considered that this adding process would help to find the larger profit values, however, we found that it didn't bring any difference to the results.

Mutation was same as the second trial.

The result is shown on Fig. 5. By the figure, we see some bias toward knapsack 1 although the code was written without any bias. After 500 generations, the solution did not improve much. Therefore, we conclude that this trial could fix the problem on heredity, but didn't find the better Pareto frontier.

4.6 Fifth Trial

On the previous trial, we tried to solve the problem with respect to the heredity. Moreover, we attempted to achieve the better Pareto frontier this time.

Up till now, for crossover, we had used only pairs of children (new pairs of individuals) for the next generation. This time, we kept both pairs of parents and pairs of children, and selected better children, that is, parents for the next generation with higher profits and smaller weights. For careful observation on this effect, we concentrated on crossover only this time; i.e., the crossover probability was 1.0 over all generations.

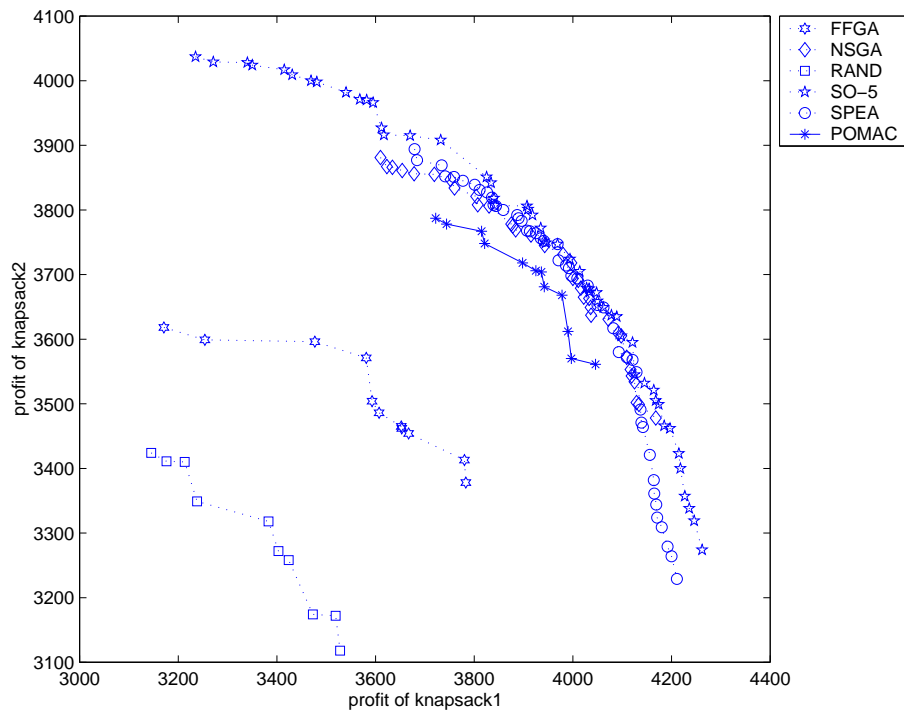


Figure 6: Result of POMAC's fifth trial (section 4.6) with other Pareto frontiers by other methods at 500th generation. This experiment improved both the number of frontier members, and reduced distance of the frontier from that achieved by SPEA.

After exchanging values in the same manner as the previous trial, both the parents and the children were stored, then population with the same size as current one was chosen for the next generation. The selection method was, first, dividing the current parents and children into two sets:

- D : simulations dominated by another simulation;
- N : remaining simulations;

and then,

$$\left. \begin{array}{l} \\ \\ \\ \end{array} \right\} |N| \begin{array}{l} = \text{ \# of required parents} \implies \text{ accept all the simulations in } N \\ < \implies \text{ accept all the simulations in } N \\ > \implies \text{ select the same number of} \end{array}$$

for the next generations;
 for the next generations, and
 choose shortages from D ; the
 larger value of r has, the
 higher priority a simulation has;
 simulations as the parents from
 N ; the larger value of r has,
 the higher priority a simulation
 has,

where

$$r = \min_{i=1,2} \frac{\text{total profit in knapsack } i}{\text{total weight in knapsack } i}.$$

As introduced in Zitzler [5], items with larger ratio of total profit to total weight are preferable. Ratio r above was defined in order to reflect this issue.

The result is shown on Fig. 6. This experiment improved the number of frontier members, and reduced distance of the frontier from that achieved by SPEA. Therefore, the way of choosing the parents for the next generation was fairly effective.

4.7 Sixth Trial

To see the effect of the rate considered on the previous trial, we changed the definition of r this time. All the procedures remained the same as the previous method. The new definition of r was

$$r = \sum_{i=1,2} (\text{total profit in knapsack } i).$$

The result was not as good as that of the previous one, which showed that considering only total profits was not sufficient, and that the relation between weight and profit played an essential role. These facts support the idea brought by Zitzler [5].

4.8 Seventh Trial

Actually, r considered on the fifth trial had a problem, because of the following reason: if, for example,

	knapsack 1	knapsack 2
ratio of parent 1:	1.0	2.0
ratio of parent 2:	1.0	1.0

and r is defined as for the fifth trial, then the parent 1 and 2 have the same possibility to be chosen. The parent 1, however, should be chosen since larger ratios are preferable. To avoid choosing wrong parents, this time, we considered the summation of ratio over the number of knapsacks, instead of the minimum. This new r could resolve the problem we had.

All the procedures for crossover are same as the fifth trial, and the new definition of r is

$$r = \sum_{i=1,2} \frac{\text{total profit in knapsack } i}{\text{total weight in knapsack } i}.$$

We also used mutation. For previous cases, we first tried to add an item, selected at random from those that were not in the knapsacks. If this broke the weight constraints, then we tried to replace an item or to remove items

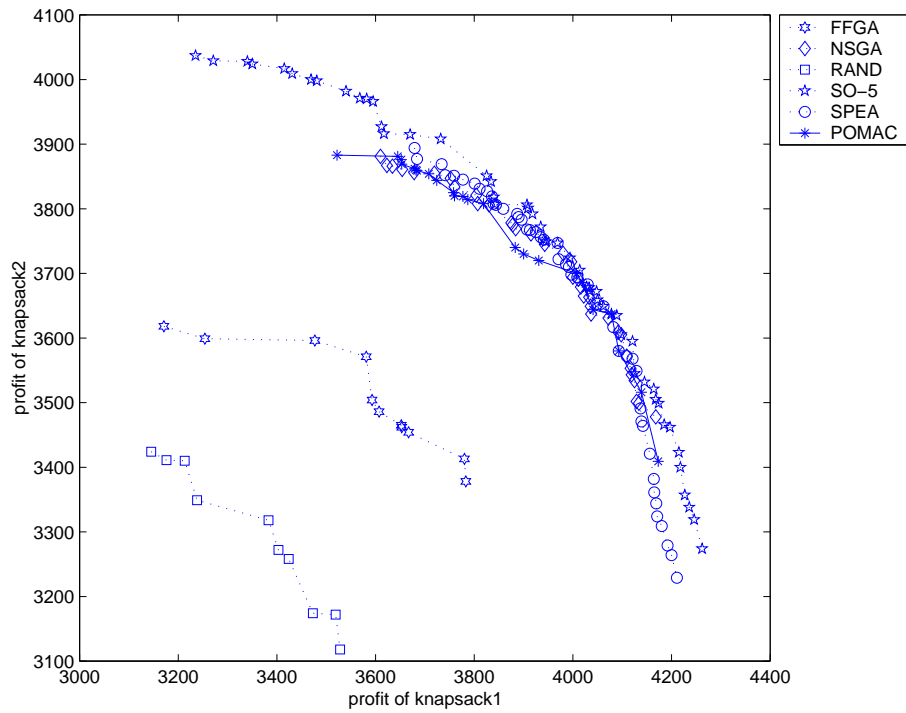


Figure 7: Result of POMAC's seventh trial (section 4.8) with the Pareto frontiers of other methods at 500th generation. The crossover probability is 0.33 and the initial number of items to be added in mutation is 10. The population size is 100 this time. Finally, the POMAC Pareto frontier reached to the SPEA's.

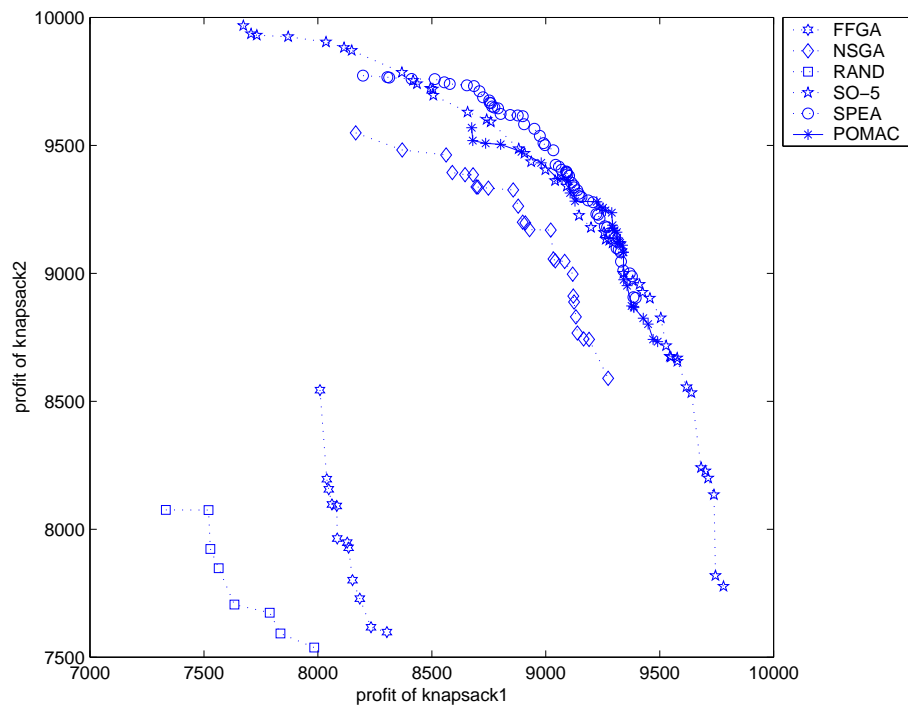


Figure 8: Frontiers by POMAC's seventh trial (section 4.8) and by other methods of the 250 item 2 knapsack problem at 500th generation. Each method was executed with population size 150.

so that the constraints could be satisfied.

Mutation for this trial was executed in the same manner, but, for early generations, we added more items, and then removed an available item having the smallest value of ratio of *profit* to *weight* to satisfy the weight constraints. We set the number of items to be added decreasing linearly to 1 over generation. We tried 10, 20 and 30 as the initial value.

In this case, crossover was more stable than mutation, because the change of values were smaller. Therefore, crossover was more essential for later generations. We set the first crossover probability at 0, 0.33 and 0.67, and increased linearly to 1.0 for each case.

We experienced several combinations of the number of items to be added in mutation and crossover probability rate. The results, however, did not show much difference. We also found that increasing the number of generation did not help to obtain better results.

Fig. 7 is the result with first crossover probability 0.33, initial number of items to be added in mutation 10. Since we obtained the reasonable result, we increased the population size to 100 from 50, about the same for other methods. (We say "about," because their population size are not mentioned exactly.) The Pareto frontier reached to that of SPEA. Compared to SPEA, the frontier was shorter in the direction to the bigger value of the total profit in *knapsack 2*, but performed better to the direction to the smaller values. We increased the population number to 150, but this trend did not change although the range of the frontier expanded a little more in both directions.

We also tried the 250 item 2 knapsack problem with 150 populations. The frontier at 500th generation is shown on Fig. 8 with ones of other methods. For POMAC, the first crossover probability 0.33 and initial number of items to be added in mutation 25 were chosen.

5 POMAC's Qualitative Comparison to the SPEA and NSGA

To compare the results by POMAC (seventh trial) to ones by other methods qualitatively, we now consider the C -measure introduced by Zitzler [5]:

$$C(X', X'') = \frac{|\{a'' \in X'' \mid \exists a' \in X' \text{ such that } a' \succeq a''\}|}{|X''|},$$

where $a \succeq b$ if and only if a dominates b or all the objective function values at a are equal to ones at b .

The value of $C(X', X'') = 1$ means that all points in X'' are dominated by or equal to points in X' . The opposite $C(X', X'') = 0$ represents the situation when none of the points in X'' are covered by the set X' .

We calculated the values of C among NSGA, SPEA and POMAC. In Zitzler [5], it was concluded that SPEA and NSGA performed better than other EAs. These algorithms were executed 30 times over 500 generations, and all 30 results for each method were published on the web (www.tik.ee.ethz.ch/~zitzler/testdata.html). For POMAC, we executed 10 times over 500 generations with population size 100, initial crossover probability 0.33 and initial number of items to be added in mutation 10. The following is the minimum, mean and maximum of the C values for each combination of the POMAC, SPEA and NSGA. For uniform results, we truncated 25% of the top and bottom data of C values:

	min	mean	max
$C(\text{NSGA}, \text{SPEA}) :$	0.090909	0.165809	0.240741
$C(\text{SPEA}, \text{NSGA}) :$	0.600000	0.727817	0.838710
$C(\text{NSGA}, \text{POMAC}) :$	0.400000	0.513959	0.631579
$C(\text{POMAC}, \text{NSGA}) :$	0.189189	0.272207	0.371429
$C(\text{SPEA}, \text{POMAC}) :$	0.692308	0.784373	0.863636
$C(\text{POMAC}, \text{SPEA}) :$	0.062500	0.106866	0.166667

We see the mean, SPEA covers both NSGA and POMAC more than 70%; while both NSGA and POMAC cover SPEA less than 20%. Comparing POMAC to NSGA, NSGA covers POMAC more than 50%; POMAC, however,

covers the NSGA less than 30%. Therefore, even though POMAC’s frontier reached to that of NSGA and SPEA (Fig. 7), the seventh trial is still not as good as NSGA and SPEA with respect to the C -measure. One of the reasons for this is that the number of populations comprising the POMAC frontier is much smaller than those of the NSGA and SPEA, and it may be possible to improve POMAC’s position by increasing the population size.

Next, we consider an S -measure, which is also introduced in Zitzler [5]. In the two-dimension case,

$$S(X) = \bigcup_{g=1}^p A_g,$$

where p is the number of populations, and A_g is the rectangular area defined by the points $(0, 0)$ and $(\text{total profit in knapsack 1}, \text{total profit in knapsack 2})$ for population p , and the axes.

Similarly to C , for uniformity, we used 50% of the data to calculate the values of S for NSGA, SPEA and POMAC:

	min	mean	max
$S(\text{NSGA}) :$	1.607×10^7	1.611×10^7	1.617×10^7
$S(\text{SPEA}) :$	1.628×10^7	1.639×10^7	1.657×10^7
$S(\text{POMAC}) :$	1.593×10^7	1.602×10^7	1.613×10^7

The reason why the S value for POMAC is smaller than that for the other two algorithms is that the frontier ranges are smaller. As we mention in the next section, modifying the way of fitness calculation may lead to a breakthrough of this situation.

Considering the results of C and S values, we can conclude that the seventh trial for POMAC found a reasonable frontier, however, was not as good as that of NSGA and SPEA qualitatively.

6 Conclusion

To modify the POMAC software to apply to the multiobjective 0/1 knapsack problem, we took the weight constraints into account. That was the

essential issue here. Although we improved the result, we failed to achieve the result that was as good as that of SPEA. This is probably not because the POMAC is unsuitable software to find the Pareto optimal solutions to the knapsack problem.

After breeding, we had to modify some of the parents for next generation to meet the weight constraints, which could ruin the inheritance system. To avoid this, it's better to take the weight constraints into account before breeding. If we could determine the fitness value which was connected to the weight constraints, then inheritance, which was an essential role of this algorithm, might work as desired.

This, however, is not easy since, after breeding, newly generated parents would tend to break the weight constraints. If we would determine the fitness value which could guarantee to keep the individual to fit the knapsacks, then we could expect to achieve the POMAC Pareto frontier of the knapsack problem which is as good as that by NSGA or SPEA.

References

- [1] Fonseca, C. M., P. J. Fleming. 1993. *Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization*. Forrest, S., editor, Proceedings of the Fifth International Conference on Genetic Algorithms, p.p. 416-423, Morgan Kaufmann, San Mateo, California.
- [2] Ford, E. David, M. Turley, J. H. Reynolds. 2000. *The Pareto Optimal Model Assessment Cycle Using Evolutionary Computation*. National Research Center for Statistics and the Environment.
- [3] Reynolds, Joel H., E. David Ford, M. Turley. 1999. *Multi-Criteria Assessment of Ecology Process Models*. Ecology, 80(2): 538-553.
- [4] Srinivas, N., K. Deb. 1995. *Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms*. Evolutionary Computation, 2(3): 221-248.
- [5] Zitzler, E., L. Thiele. 1999. *Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach*. IEEE Transactions on Evolutionary Computation, 3(4): 257-271.

