Mathematical & Computational Applications, Vol. 8, No. 1 , pp. 35-47, 2003

DESIGN-OPTIMIZATION OF MECHANICAL COMPONENTS USING THE "MULTIPLE OBJECTIVE TABU SEARCH" METHOD

Adil Baykasoğlu

University of Gaziantep, Department of Industrial Engineering, 27310 Gaziantep, TURKEY
baykasoglu@gantep.edu.tr

Abstract- The multiple objective version of the tabu search (TS) algorithm was initially developed by Baykasoglu *et al* [1-3]. The idea of applying tabu search to multiple objective optimization, inspired from its solution structure, in which tabu search works with more than one solution (neighborhood solutions) at a time. This situation creates the opportunity to evaluate multiple objectives simultaneously in one run. To enable the original tabu search algorithm to work with more than one objective the selection and updating stages were redefined. Other stages are identical to the original tabu search algorithm. In this paper, multiple objective tabu search algorithm is used to solve mechanical component design problems with multiple objectives. Although there exists a number of classical techniques, meta-heuristic algorithms including TS have an edge over the classical methods in that they can find multiple Pareto-optimal solutions in one single run. In the paper four mechanical component design problems borrowed from the literature are solved. The results are compared with several other solution techniques including multiple objective genetic algorithms. It is observed that in many of the cases the multiple objective tabu search algorithm can find better and much wider spread of solutions than the compared algorithms.

Key Words- Multiple objective tabu search, mechanical design, optimization

1. INTRODUCTION

Many engineering design problems involve multiple objectives. In operational research literature these problems are known as 'Multiple Objective Optimization' (MOO) problems. Generally a MOO problem is of the following form:

$$\min \ or \ \max \ F(X)$$
$$such\ that; \qquad (1)$$
$$X \in S = [X | X \in A^n, g_i(X) \le a_i,\ h_j(X) = b_j] \qquad i=1,2,...,m,\ j=1,2,...,n$$

where: X is an n-dimensional vector of the decision variables; $F(X)=\{f_1(X), f_2(X),..., f_k(X)\}$ is the set of objective functions; and S is the set of feasible solutions, bounded by m inequality constraints (g_i) and n equality (h_j) constraints, a_i and b_j are constants. For continuous variables $A=\Re$, for discrete variables A contains set of permissible values.

In MOO problems including engineering design problems objectives are generally conflicting objectives. In general, existence of conflicting objectives results in a number of optimal solutions. The concept of Pareto optimality is generally used to characterize optimal solutions to a MOO problem. The Pareto-optimal (non-dominated) solution is defined as follows: a solution $X^* \in S$ is Pareto optimal if and only if there exists no $X \in S$ such that $f_i(X) \le f_i(X^*)$ for $i=1,2,3,...,k$ with $f_i(X) < f_i(X^*)$ for at least one value of i. In other words, the solution X^* is Pareto optimal if no objective function can be improved without worsening at least one other objective function. Since no one

Pareto-optimal solution can be said better than another without further consideration, it is desired to find as many such Pareto-optimal solutions as possible.

Unfortunately, classical techniques impose several limitations on solving mathematical programming models including the multiple objective ones [1-3]. The problem is mainly related to inherent solution mechanisms of these techniques. Their solution strategies are generally depended on the type of objective and constraint functions (linear, non-linear etc.) and the type of variables used in the problem modeling (integer, real etc.)[3]. Their efficiency is also dependent on the size of the solution space, number of variables and constraints used in the problem modeling, and the structure of the solution space (convex, non-convex, etc.) [1,2]. They also do not offer a general solution strategy that can be applied to problem formulations where, different type of variables, objective and constraint functions are used [3]. For example, *simplex Algorithm* can be used to solve models with linear objective and constraint functions; *geometric programming* can be used to solve non-linear models with a posynomial or signomial objective function etc. However, most of the engineering design problems require different types of variables, objective and constraint functions simultaneously in their formulation. Therefore, classic optimization procedures are generally not adequate or easy to use for their solution [3,4].

Researchers have spent a great deal of effort in order to adapt many engineering design problems to the classic optimization procedures. Many examples can easily be found in the literature. An interesting application from the author's previous research, which was related to cutting conditions optimization using geometric and dynamic programming might be an interesting example [5]. It is not easy to formulate a real life problem that suits a specific solution procedure. In order to achieve this, it is necessary to make some modifications and/or assumptions on the original problem parameters (rounding variables, softening constraints etc.)[3]. This certainly affects the solution quality. A new set of problem and model independent heuristic optimization techniques were proposed by researchers to overcome drawbacks of the classical optimization procedures. These techniques are efficient and flexible [3,4]. They can be modified and/or adapted to suit specific problem requirements. Three of these widely accepted and applied techniques are known as *Genetic Algorithms* [6], *Tabu Search* [7] and *Simulated Annealing* [8].

In this study the focus is on solving multiple objective design of mechanical components using the multiple objective tabu search algorithm (MOTS). MOTS algorithm was initially proposed by the author [1,3]. The idea of applying tabu search to MOO comes from its solution structure, in working with more than one solution (neighborhood solutions) at a time. In fact, any solution methodology that works with more than one solution vector at a time can be effectively used for MOO like genetic algorithms [1]. Due to its population-based search characteristic, the genetic algorithms are frequently applied to MOO problems. To enable the TS algorithm to work with more than one objective, selection and updating stages of the basic TS are redefined. Other stages are similar to the original tabu search algorithm. In contrast to original TS algorithm, MOTS algorithm has two more lists in addition to the tabu list. The first one is the *Pareto list*, which collects selected non-dominated solutions found by the algorithm. The second one is the *candidate list*, which collects all other non-dominated solutions, which are not selected as Pareto optimal solutions in the current iteration.

These solutions may become seed solutions if they maintain their non-dominated status in later iterations. The candidates list plays also an important role, it gives the opportunity to diversify the search [1,3].

In this research paper, the efficiency of MOTS is investigated in finding diverse Pareto-optimal front in a number of engineering design problems. The first design problem is the design of machine tool spindles. The second problem is gear train design problem. The third problem is I-beam design problem and the final problem is two-bar truss design problem. The quality of solutions obtained from the MOTS is compared with the results of several other techniques including genetic algorithms in the paper. It is observed that MOTS is eligible to solve engineering design problems. In most of the cases MOTS converged solutions, which are better or not worse than the solutions generated by the other algorithms.

2. AN OVERVIEW OF THE MOTS ALGORITHM

The elements of the MOTS algorithm for finding Pareto-optimal solutions in MOO problems with any type of variables (integer, zero-one, discrete or continuous) and performance functions (linear, non-linear, convex, non-convex) are defined as follows [1,3]:

Initial Solution: *A randomly generated or feasible solution vector is initial solution.*

Generation of neighborhood solutions: *To generate a neighbor for any type of variable, new values are formulated as [1-3]:*

Integer variable $$x_i^* = x_i + \text{integer}[(2*random() - 1)*stepi_i]$$

Zero-One variable $$x_i^* = \begin{cases} 1 & if\ x_i = 0 \\ 0 & if\ x_i = 1 \end{cases} \tag{2}$$

Discrete variable $$x_i^* = d_{(l+\text{integer}[(2*random(\ )-1)*stepd_i])} \quad if \quad x_i = d_l$$

Continuous variable $$x_i^* = x_i + (2 * random(\ ) - 1) * stepc_i$$

Where; x_i : Value of the i^{th} variable prior to the neighborhood move. x_i^* : Value of the i^{th} variable after the neighborhood move. *random*( ): Random number generator, where *random*( )$\in (0,1)$. $stepi_i$, $stepd_i$, $stepc_i$: Step size for integer, discrete and real variables. d_l : The l^{th} element of the discrete variable subset X^d. integer[ ]: Function to convert a real value to an integer value. According to the types of variables used in the model, the appropriate movement strategies are used to generate a previously determined number of feasible, non-tabu, neighborhood solutions from the current seed solution. Neighborhood solutions must also be non-dominated by the current seed solution.

Selection of the seed solution: Selection of the seed solution is performed using the Pareto optimality logic (domination and non-domination). Pareto optimality is an economics term for describing a solution for multiple objectives. It is generally used to characterize optimal solutions to a MOO problem. The Pareto optimal (non-dominated) solution is defined as follows: a solution $x^* \in s$ is Pareto optimal if and only if there exists no $x \in s$ such that $f_i(x) \leq f_i(x^*)$ for $i=1,2,3,\ldots,m$ with $f_i(x) < f_i(x^*)$ for at least one value of i. In other words, the solution x^* is Pareto optimal if no objective function can be improved without worsening at least one other objective function. Based on the

Pareto optimality logic, the selection of the best neighborhood solution as the new seed solution is performed in the following manner:

i) For each neighborhood solution vector, the corresponding objective function values are calculated. In the example given below, the neighborhood size is three and there are two real variables and two objective functions to be maximized.

Seed solution followed by objective function values			*3 Neighborhood solutions (non-tabu, feasible and not dominated by the seed solution)*		*Corresponding objective function values of neighborhood solutions*
			(6.3 6.1)	→	(60.08 47.09)
(4.8 4.6)	(52.4 40.93)	→	(6 6)	→	(58.79 46.54)
			(6.4 6)	→	(60.39 46.86)
Variable values	*Objective values*		*Variable values*		*Objective values*

ii) Candidate seed solutions within the neighborhood solutions are identified. Candidate seed solutions should not be dominated by other neighborhood solutions, solutions in the Pareto list or solutions in the candidate list. This process is illustrated below.

Seed solution followed by objective function values		*Neighborhood solutions (non-tabu, feasible and not dominated by the seed solution)*		*Objective function values of neighborhood solutions*
		(6.3 6.1)	→	(60.08 47.09) O
(4.8 4.6) (52.4 40.93)	→	(6 6)	→	(58.79 46.54)
		(6.4 6)	→	(60.39 46.86) O

Pareto List	Candidate List
(0 0) (0 0) ✖	(4 3) (46.93 33.98) ✖
(0.5 0.5) (16.97 13.44) ✖	(3 4) (42.64 36.93) ✖
(1 1) (24 19) ✖	
(2 2) (33.94 26.87) ✖	
(3 3) (41.57 32.91) ✖	
(3.8 3.6) (46.57 36.26) ✖	
(4.8 4.6) (52.4 40.93)	

(O: Candidate solutions, ✖: Eliminated solution from previous iterations)

iii) One of the candidate solutions is randomly selected as the new seed solution. If there are no candidate solutions in the current neighborhood, the oldest solution from the candidate list is selected as the seed solution.

It can be seen from the above selection strategy that the dominated solutions are not taken into consideration, because the purpose is to find the Pareto optimal solutio which do not dominate each other. MOTS algorithm works with two more dynamic, lists namely Pareto list and Candidate list, the Candidate list (which collects potential candidate Pareto optimal solutions and updates their status dynamically) enables the search process to avoid abandoning while searching and diversify the search (this case can also be imagined as avoiding to fall into the trap of local optima in global optimization). Pareto list collects the seed (or currently selected) potential Pareto optimal solutions and dynamically updates their status.

Updating the lists: The initial feasible solution vector is recorded as the first known Pareto solution vector. The solutions, which are dominated by any neighborhood solution, are removed from both Pareto and candidate lists in each iteration. Then the seed solution is added to the Pareto list, and other candidate solutions are put into the candidate list. This process is shown on the same example below.

Pareto List	Candidate List
(0 0) (0 0) ✖	(4 3) (46.93 33.98) ✖
(0.5 0.5) (16.97 13.44) ✖	(3 4) (42.64 36.93) ✖
(1 1) (24 19) ✖	(6.3 6.1) (60.08 47.39)
(2 2) (33.94 26.87) ✖	
(3 3) (41.57 32.91) ✖	
(3.8 3.6) (46.57 36.26) ✖	
(4.8 4.6) (52.4 40.93) ✖	

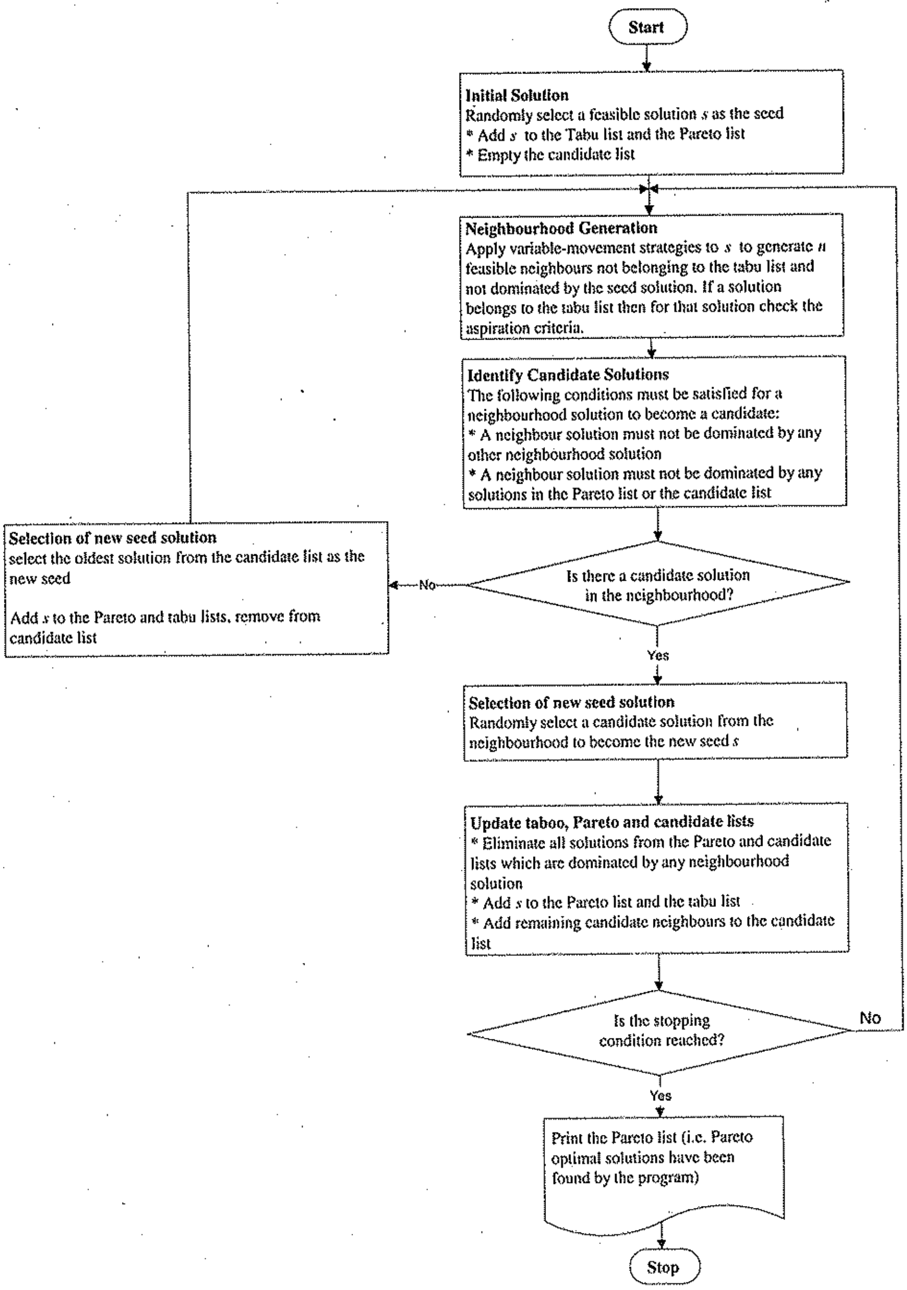

Figure 1. The flowchart of the MOTS algorithm [3]

Selected seed solutions for an arbitrarily defined number of previous moves are considered as tabu, since reusing one of them may trap the algorithm into cycling through recent, moves. In our algorithm, the tabu list contains m solutions, corresponding to the last m seed solutions. The tabu list is circular, i.e. when it is full a new item replaces the head of the list.

Aspiration criteria: In combinatorial optimization problems, solution vectors are generally generated indirectly using several features of the problem at hand. For example in a manufacturing cell formation problem a new solution can be generated by randomly reassigning part-machine pairs to different cells in each iteration in order to obtain a new solution [9]. In such a case instead of putting the whole solution vector in the tabu list as tabu solution only indices (features) of randomly selected and reassigned part-machine pairs are put into the tabu list. However, reselection of these features for new solution generation in later iterations might generate different solution vectors, as these features themselves are not the solution vectors. Therefore, optimal solutions may be missed if these features are considered strictly as tabu. In order to prevent this situation in TS applications an aspiration criterion needs to be defined to override the tabu status of features when necessary. But, if the entire solution vector is put into to the tabu list then it is not necessary to define an aspiration criterion, which is generally the case in design optimization problems.

Termination: If a previously determined number of iterations is reached, or if the candidate list is empty and the algorithm cannot find any new candidate solutions, the program terminates. The general flowchart of the algorithm is given in Figure 1 [3].

3. MULTIPLE OBJECTIVE DESIGN OPTIMIZATION PROBLEMS

MOTS algorithm is applied to four design-optimization problems collected from the literature. In each application the algorithm successfully found good solutions in comparison to the reported solutions. The MOTS algorithm is programmed using C++. The program is an object oriented one and uses advanced linked list constructs. The program is tested on a Pentium III-MMX model PC at 450 MHz (128 MB RAM).

3.1. Compound Gear Train Design

This problem is taken from Deb *et al* [10]. A compound gear train is to be designed to achieve a given gear ratio between the driver and driven shafts (see Figure 2). The objective of the gear train design problem is to find the number of teeth in each of the four gears so as to minimize:

i. The error between the obtained gear ratio and a required gear ratio of 1/6.931 [10] and

ii. The maximum size of any of any of the four gears.

Since the number of teeth is integer, all four variables must be integer. By denoting the variable vector $x = (x_1, x_2, x_3, x_4) = (T_d, T_b, T_a, T_f)$, two objective optimization problem is given as follows:

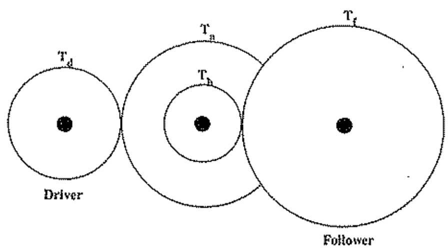

Figure 2. Compound gear train

$$
\begin{aligned}
&\text{minimise} && \left[\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right]^2 \\
&\text{minimise} && x = \max(x_1, x_2, x_3, x_4) \qquad\qquad (3) \\
&s.t. && 12 \le x_1, x_2, x_3, x_4 \le 60 \\
& && x_1, x_2, x_3, x_4 \ge 0 \quad \& \ \mathit{Integer}
\end{aligned}
$$

For solving the compound gear train design problem, MOTS parameters are set as follows; neighborhood size =150, tabu list size =20, step sizes for the fist, second, third and fourth variables are taken as 5, 5, 5, 5, maximum number of iterations is set to 5000. Using this parameter set MOTS found 41 solutions after 402 iterations in 2 seconds. All obtained solutions are shown in Figure 3. The MOTS is also run with different parameter sets and no significant differences observed on the solution quality. The comparison of the best values of objectives obtained from MOTS and other techniques presents that solutions of MOTS are not dominated. Moreover, MOTS found more Pareto-optimal solutions than all other techniques, and the spread of the MOTS solutions are much wider in Pareto-optimal front see reference [10]. The comparisons are also summarized in Table 1.

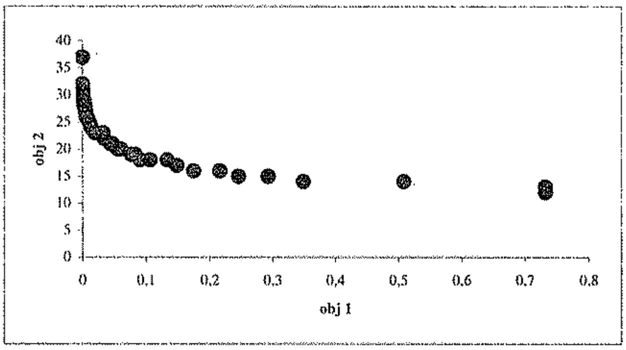

Figure 3. MOTS solutions for compound gear train problem

Table 1. Comparison of solutions for the compound gear train design problem

	Number of Pareto-optimal solutions found	*Best objective function values found for obj-1 and obj-2and solution quality records*	*Comput. time (in sec)*
MOTS	**41**	{**$1.83*10^{-8}$**, 37} {0,732258, **12**}	2
GA-1 of Deb et al [10]	6	{**$1.83*10^{-8}$**, 37}{$2.47*10^{-4}$, **30**}	Not available
GA-2 of Deb et al [10]	23	Dominated by MOTS and GA-2	Not available
Lagrangian [10]	1	Not Dominated by any other technique	Not available
Branch-and-Bound [10]	1	Dominated by MOTS	Not available
Single Obj GA-1 [10]	1	Not Dominated by any other technique	Not available
Single Obj GA-2 [10]	1	Not Dominated by any other technique	Not available

3.2. Two Bar Truss Design

This problem is taken from Deb *et al* [10]. The problem was originally studied using the $\in$-constraint method. The truss has to carry a certain load without elastic failure (see Figure 4). Thus, in addition to the objective of designing the truss with minimum volume, there are additional objectives of minimizing stresses in each of the two members AC and BC. The following two-objective optimization problem for three variables y (vertical distance between B and C in m), x_1 (length of AC in m) and x_2 (length of BC in m) is constructed by Deb *et al* [10].

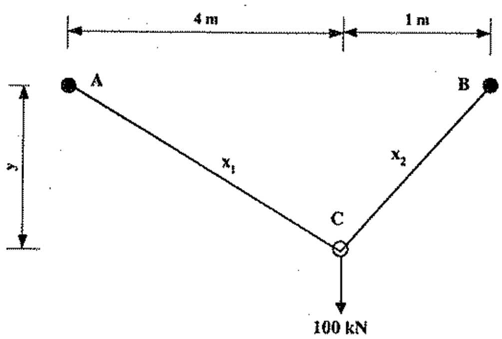

Figure 4. Two-bar truss

$$
\begin{aligned}
&\text{minimise} && x_1\sqrt{16+y^2}+x_2\sqrt{1+y^2} \\
&\text{minimise} && \max(\sigma_{AC},\sigma_{BC}) \\
&s.t. && \max(\sigma_{AC},\sigma_{BC}) \le 1*10^5 \\
& && 1 \le y \le 3 \\
& && x_1, x_2, y \ge 0 \quad \& \ continuous
\end{aligned}
\tag{4}
$$

$$
\sigma_{AC} = \frac{20\sqrt{16+y^2}}{yx_1}, \quad \sigma_{BC} = \frac{80\sqrt{1+y^2}}{yx_2}
$$

For solving the compound gear train design problem, MOTS parameters are set as follows; neighborhood size =150, tabu list size =20, step sizes for the fist, second, and third variables are taken as 2, 0.01, 0.01, maximum number of iterations is set to 5000. Using this parameter set MOTS found 651 solutions after 1562 iterations in 18 seconds. All obtained solutions are shown in Figure 5. The MOTS is also run with different parameter sets and no significant differences observed on the solution quality. The comparison of the best values of objectives obtained from MOTS and other techniques presents that solutions of MOTS are not dominated. Moreover, MOTS found much more Pareto-optimal solutions than the other techniques ($\in$-constraint method and Deb *et al*'s GA-1 and GA-2 [10]), and the spread of the MOTS solutions are much wider than the $\in$-constraint method see reference [10]. The solutions of MOTS are spread in the following range {0.005902 m^3, 99557 kPa}, {0.056623 m^3, 8432 kPa}, whereas the spread of solutions of Deb *et al*'s GA-2 [10] are in the {0.00407 m^3, 99755 kPa}, {0.05304 m^3, 8439 kPa} range. If the minimization of the stress is important, MOTS finds a solution with stress as low as 8432 kPa, whereas the GA-2 of Deb *et al* [10] has found 8439 kPa, $\in$-constraint method has found 83268 kPa. As a result, the quality of solution obtained by MOTS and GA-2 of Deb are nearly equal. But MOTS found much more Pareto-optimal solutions in the Pareto-optimal front. Both MOTS and GA-2 dominated $\in$-constraint method in terms of solution quality and number of Pareto-optimal solutions found.

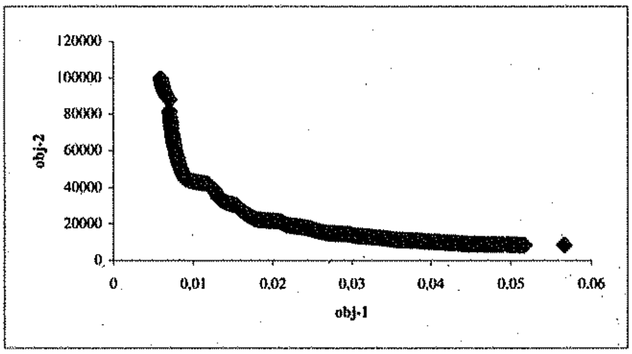

Figure 5. MOTS solutions for two-bar truss problem

3.3. Machine Tool Spindle Design

The machine tool spindle problem (see Figure 6) is originally modeled by Eschenauer *et al* [11]. Coello [12] remodeled this problem as a MOO problem and solved it using their genetic algorithm, which is known as MOSES. They also compared their results with four other MOO techniques with respect to best results obtained for each objective function. In this test study, their model is solved by MOTS. The model is given in Equation 5. More details about the model can be obtained from Coello [12].

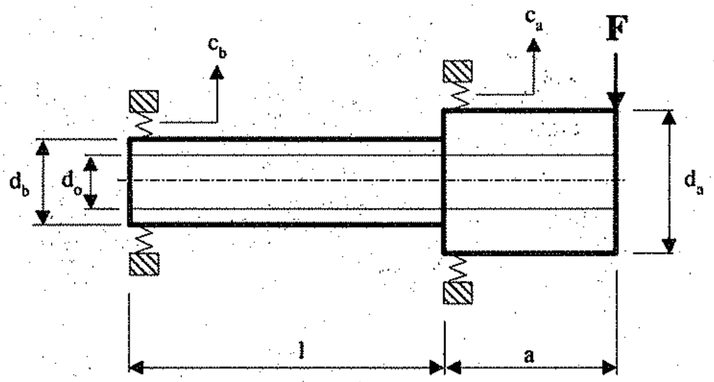

Figure 6. Sketch of the machine tool spindle

minimise $\frac{\pi}{4}\left[a(d_a^2 - d_o^2) + l(d_b^2 - d_o^2)\right]$

minimise $\frac{Fa^3}{3EI_a}(1 + \frac{l}{a}\frac{I_a}{I_b}) + \frac{F}{c_a}\left[(1 + \frac{a}{l})^2 + \frac{c_a a^2}{c_b l^2}\right]$

$I_a = 0.049(d_a^4 - d_0^4)$, $I_b = 0.049(d_b^4 - d_0^4)$, $c_a = 35400|\delta_{ra}|^{1/9} d_a^{10/9}$, $c_b = 35400|\delta_{rb}|^{1/9} d_b^{10/9}$

s.t. $l - l_g \le 0$

$l_k - l \le 0$

$d_{a1} - d_a \le 0$

$d_a - d_{a2} \le 0$ 5

$d_{b1} - d_b \le 0$

$d_b - d_{b2} \le 0$

$d_{om} - d_o \le 0$

$p_1 d_o - d_b \le 0$

$p_2 d_b - d_a \le 0$

$\left|\Delta_a + (\Delta_a - \Delta_b)\frac{a}{l}\right| - \Delta \le 0$

$d_o, l \ge 0$, *continuous* & d_a, d_b *discrete*

In the machine spindle design model d_a and d_b are discrete variables and d_a should be selected from the following set {80, 85, 90, 95}, and d_b from the set {75, 80, 85, 90}. For solving the spindle design problem, MOTS parameters are set as follows; neighborhood size =10, tabu list size =20, step sizes for the fist, second, third and fourth variables are taken as 5, 5, 4, 4, maximum number of iterations is set to 1000. Using this parameter set MOTS found 128 solutions after 348 iterations in 9 seconds. All obtained solutions are shown in Figure 7. The MOTS is also run with different parameter sets and no significant differences observed on the solution quality. The comparison of the best values of objectives obtained from MOTS and other techniques reported in Coello [12] presents that solutions of MOTS are not dominated. These comparisons are also shown in Table 2.

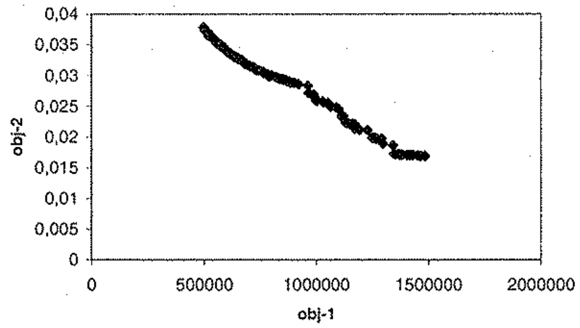

Figure 7. MOTS solutions for machine spindle design problem

Table 2. Comparison of the best results for each objective of the machine spindle design problem (For each method the best results for $f_1(x)$ and $f_2(x)$are shown in boldface)

Techniques	d_o	l	d_a	d_b	$f_1(x)$	$f_2(x)$
Monte Carlo-1	59.08	189.17	90	75	**606765.47**	0.032463
Monte Carlo-2	26.26	193.29	90	85	1457748.36	**0.019242**
GA(binary)	60.00	200.00	80	75	**494015.44**	0.038087
GA(binary)	25.00	190.09	95	90	1643777.68	**0.016613**
GA(floating point)	56.16	194.49	95	90	**1124409.37**	0.017951
GA(floating point)	25.35	189.58	95	90	1637052.38	**0.016615**
Literature	63.89	183.29	85	80	**531183.70**	0.030215
Literature	66.45	183.36	95	85	694200.03	**0.023101**
MOTS	59.84	199.26	80	75	**497644.1**	0.037839
MOTS	39.02	199.62	85	80	1485169	**0.016894**

3.4. I-Beam Design

In Figure 8, design optimization for a simply supported I-beam is shown. Osyczka [13] originally modeled this problem. Coello and Christiansen [14] remodeled the problem as MOO problem and solved the model using genetic algorithm, which is known as MOSES. They also made an extensive comparison of their results with some other MOO techniques with respect to best results obtained for each objective function. In this study, their model is solved by MOTS. The model is given in Equation 6. More details about the model can be obtained from Coello and Christiansen [14].

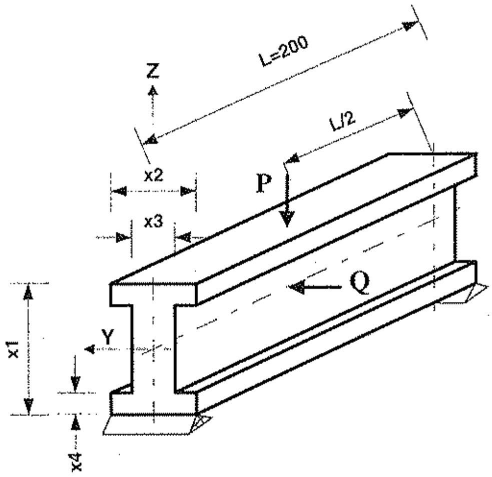

Figure 8. Sketch of the simply supported I-beam

$$
\begin{aligned}
&\text{minimise} && 2x_2x_4 + x_3(x_1 - 2x_4) \\
&\text{minimise} && \frac{60000}{x_3(x_1-2x_4)^3 + 2x_2x_4(4x_4^2 + 3x_1(x_1-2x_4))} \\
&s.t. && 16 - \frac{180000x_1}{x_3(x_1-2x_4)^3 + 2x_2x_4(4x_4^2+3x_1(x_1-2x_4))} - \frac{15000x_2}{(x_1-2x_4)x_3^3 + 2x_4x_2^3} \geq 0 \\
& && 10 \leq x_1 \leq 80 \\
& && 10 \leq x_2 \leq 50 \\
& && 0.9 \leq x_3 \leq 5 \\
& && 0.9 \leq x_4 \leq 5 \\
& && x_1, x_2, x_3, x_4 \geq 0 \; (continuous)
\end{aligned}
\qquad (6)
$$

For solving the I-beam design problem MOTS parameters are set as follows; neighborhood size =10, tabu list size =20, step sizes for the fist, second, third and fourth variables are taken as 5, 5, 2, 2, maximum number of iterations is set to 1000. Using this parameter set MOTS found 92 solutions after 178 iterations in 8 seconds. All obtained solutions are shown in Figure 9. The MOTS is also run with different parameter sets and no significant difference is observed on the solution quality. The comparison of the best values of objectives obtained from MOTS and other techniques reported in Coello and Christiansen [14] presents that solutions obtained from the MOTS are not dominated.

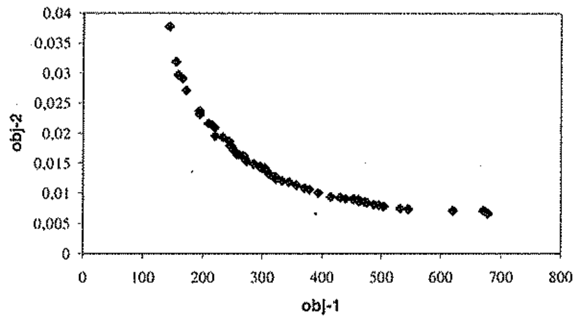

Figure 9. MOTS solutions for I-beam design problem

4. CONCLUSIONS

Any optimization technique, which works with more than one solution vector in its inherent solution mechanism, such as TS, can be effectively used for solving MOO models including engineering design ones. TS is a heuristic neighborhood search algorithm that works with a set of potential solutions known as neighborhood solutions. This gives TS an advantage in solving MOO problems directly without requiring additional method. Based on this observation, recently Baykasoglu *et al* [1-3] develope the multiple objective versions of TS, which is known as MOTS.

In this study an application of MOTS to solve the mechanical components design models is presented. Four different problems collected from the literature are solved. The first problem is compound gear train design problem, the second one is two-bar truss design, the third one is machine tool spindle design and the final problem is I-beam design problem. In all of the design problems MOTS has found very good results. The solutions obtained from MOTS dominated solutions obtained from the classical techniques and are better than the compared genetic algorithm solutions in terms of number of Pareto-optimal solutions obtained. It can finally be concluded that MOTS is a viable candidate for solving engineering design problems.

REFERENCES

1. A. Baykasoğlu, S. Owen, and N. Gindy, A taboo search based approach to find the Pareto optimal set in multiple objective optimization, *Eng. Opt.*, **31**, 731-748,1999.
2. A. Baykasoğlu, S. Owen, and N. Gindy, Solution of goal programming models using a basic taboo search algorithm, *J. Opl. Res*, **50**, 960-973, 1999.
3. A. Baykasoğlu, Goal programming using the multiple objective tabu search, *J. Opl. Res*, **52**, 1359-1369, 2001.

4. A. K. Dhingra, B. H. Lee, A genetic algorithm approach to single and multiobjective structural optimization with discrete-continuous variables. *Int. J. for Num. Meth. in Engineering,* **37**, 4059-4080, 1994.
5. A. I. Sonmez, A. Baykasoğlu, T. Dereli, I. H. Filiz, Dynamic optimization of multipass milling operations via geometric programming, *Int. J. of Machine Tools and Manufacture,* **39**, 297-320, 1999.
6. J. H. Holland (1992). Genetic algorithms, *Scientific American,* 44-50, 1992.
7. F. Glover, Tabu search: a tutorial, *Interfaces,* **20**, 74-94, 1990.
8. S. Kirkpatrick, C. D. Gelatt Jr, M. P. Vecchi, Optimisation by simulated annealing. *Science*, **220,** 671-680, 1983.
9. A. Baykasoğlu A, N. N. Z. Gindy, MOCACEF 1.0: Capability based approach to form part-machine groups for cellular manufacturing applications. *Int. J. of Prod. Res.*, **38**,1133-1161, 2000.
10. K. Deb, A. Patrap, and S. Moitra, Mechanical Component Design for multi-objective using Elitist non-dominated sorting GA, KanGAL report 200002, Indian Institute of Technology, Kanpur, India, 2000.
11. H. Eschenauer, J. Koski, A. Osyczka, *Multicriteria Design Optimization.* Springer-Verlag, 1990.
12. C. C. A. Coello, *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design*, PhD thesis, Department of Computer Science, Tulane University, New Orleans, LA, 1996.
13. A. Osyczka, *Multicriteria Optimization for Engineering Design.* In Gero JS, editor, Design Optimization, pages 193-227. Academic Press, 1985.
14. C. C. A. Coello, A. D. Christiansen, MOSES: A multiple objective optimization tool for engineering design. *Eng. Opt.*, **31**, 337-368, 1999.