Proceedings of DETC' 03
ASME Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
Chicago, Illinois, September 2-6, 2003

DETC2003/DAC-48750

DESIGN OPTIMIZATION OF A VEHICLE B-PILLAR SUBJECTED TO ROOF CRUSH USING MIXED REACTIVE TABOO SEARCH

Karim Hamza
Ph.D. Pre-Candidate, Mechanical Engineering Dept., University of Michigan, Ann Arbor, MI 48109-2125
khamza@engin.umich.edu

Kazuhiro Saitou[*]
Assistant Professor, Mechanical Engineering Dept., University of Michigan, Ann Arbor, MI 48109-2125
kazu@engin.umich.edu

Ashraf Nassef
Assistant Professor, Mechanical Engineering Dept., American University in Cairo
nassef@aucegypt.edu

ABSTRACT

The primary obstacle in automated design for crashworthiness is the heavy computational resources required during the optimization processes. Hence it is desirable to develop efficient optimization algorithms capable of finding good solutions without requiring too many model simulations. This paper presents an efficient mixed discrete and continuous optimization algorithm, Mixed Reactive Taboo Search (MRTS), and its application to the design of a vehicle B-Pillar subjected to roof crush conditions. The problem is sophisticated enough to explore the MRTS' capability of identifying multiple local optima with a single optimization run, yet the associated finite element model (FEM) is not too large to make the computational resources required for global optimization prohibitive. The optimization results demonstrated that a single run of MRTS identified a set of better designs with smaller number of simulation runs, than multiple runs of Sequential Quadratic Programming (SQP) with several starting points.

INTRODUCTION

According the National Highway Traffic Safety Administration, there were over six million vehicle crashes in the United States in the year 2000 alone, which claimed the life of more than forty thousand people and resulted in losses of more than twenty seven billion dollars [1]. It is one of the prime duties of vehicle designers to ensure that their vehicles meet governmental safety regulations. Additionally, higher levels of safety, while keeping the structural mass and production cost minimal, is important for increasing the vehicle attractiveness in a highly competitive market.

Despite the recent advances in computer aided design (CAD) tools, crashworthiness design of vehicle structures still remains, for the most part, an intuition driven procedure. Driven by past experience, a designer assumes a design, uses CAD software or field crash experiments for testing the design then re-iterates based on intuition gained from the tests. A potentially more effective approach would be to use CAD models along with optimization techniques to automatically improve the design. However, the main obstacles to automated crashworthiness optimization are the requirement of massive computational resources for simulation of full vehicles and the existence of simulation errors that present noise to the optimization models.

This paper presents an efficient optimization technique, the Mixed Reactive Taboo Search (MRTS), and its application to the design of a vehicle B-Pillar (Fig. 1) subjected to roof crush conditions. MRTS is an updated blend of Reactive Taboo Search (RTS) for discrete optimization [2] and Continuous Reactive Taboo Search (CRTS) for continuous optimization [3]. The design of the B-Pillar is formulated as a mixed discrete and continuous optimization problem. The problem is sophisticated enough to explore the MRTS' capability of identifying multiple local optima with a single optimization run, yet the associated finite element model (FEM) is not too large to make the computational resources required for global optimization prohibitive. It is demonstrated that a single run of MRTS identified a set of better designs with smaller number of simulation runs, than multiple runs of Sequential Quadratic Programming (SQP) with several starting points.

The paper starts with a brief review of the relevant literature followed by a detailed description of the B-Pillar Optimization problem. Then, the algorithm of MRTS is explained and the results of its application to the B-Pillar optimization compared with multi-start SQP are presented. The paper ends with a general discussion.

RELVANT LITERATURE

Most of the recent successful attempts for automating crashworthiness design optimization generally fall into one of

[*] corresponding author

three categories: i) topology optimization, ii) parametric design optimization and iii) development of crashworthiness models.

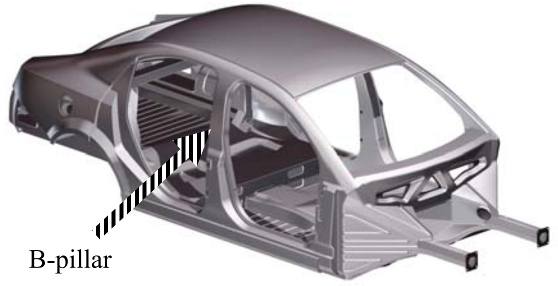

Fig. 1. B-pillar in vehicle body (Source: American Iron and Steel Institute: http://www.steel.org/).

Examples of topology optimization can be found in [4-8], however, the main use of topology optimization is during the conceptual design stage, where the exact dimensions and specifications of the vehicle are not yet finalized and it is required to find a good overall layout and/or connectivity of the structure members. Topology optimization rarely produces final dimensions of structural members that are to be manufactured. Parametric design optimization, on the other hand is of use in the final design stages, as it assumes the overall geometry, shapes and connectivity to be finalized and sets forth to find the dimensions and member cross-sections that better serve the design objective. As a drawback however, parametric design requires more detail and accuracy in the optimization model than topology optimization does.

Development of approximate models for simulating the vehicle behavior during crash, yet do not require the larger computational resources as those required for full FEM simulations, is also an active area of research. On one side, there are the reduced and lumped parameter models [9,10,11] based on simplified physical representations of the vehicle, while on the other hand, there are the surrogate modeling techniques which are mathematical methods for general purpose functional approximation [12-16]. While lumped parameter models have the advantage of including an embedded physical model, they are generally not very useful in parametric design optimization because of their lack of detail.

Although surrogate models abstract the physics of the problem, on the other hand, they have the advantage of associating the vehicle model output with (in theory) any desired level of detail for the input design variables, and they also smooth out much of the noise in the objective function and constraints that arises from errors in FEM simulations. Currently, surrogate modeling dominates crashworthiness parametric design optimization and has been effectively applied to partial vehicle structures [17,18,19] as well as full vehicles [20].

Aside from abstracting the physical model of the vehicle however, a main drawback of surrogate modeling is the computational effort for constructing the surrogate model. Optimization with surrogate models can lead to near-global optima, provided the data used in constructing the model has enough spread and accuracy over the search space. In practice however, constructing a surrogate model that has *both* wide spread and good accuracy becomes computationally very expensive as the number of design variables increases. Surrogate models that are constructed within a confined region of the search space often produce optimization results that are local to such region as in the study performed in [20].

Parametric design optimization for crashworthiness of full vehicles requires massive computational resources that are often too prohibitive. For example, the research conducted by Yang *et. al.* [20] required 512 processors running for 72 hours to perform only two local optimization steps. Thus, global optimization of full vehicles is yet out of scope. However, in some cases, it is possible to break down the full vehicle structure into smaller subsystems that represent small to medium sized optimization problems that can be more efficiently handled.

An example of such subsystems is the one considered in this paper, where the B-Pillar of the vehicle is the primary load carrying member during roof crush and thus the load carrying requirement of the pillar can be estimated and set as a required performance. Being reasonable in terms of requirement of computational resources allows for an efficient optimization technique such as MRTS, which is presented in this paper to perform global optimization using the full FEM model of the subsystem, without the need for approximations using surrogate models.

B-PILLAR ROOF CRUSH MODEL

The design problem under consideration is that of a B-Pillar of a mid-sized car subjected to roof crush. The required performance of the pillar according to a federal regulation is that it should be able to support a quasi-static load equal to some multiple of the vehicle weight, without getting deformed beyond a certain amount, since the deformation represents intrusion into the passenger cabin.

A finite element model of a generic B-Pillar is shown in Fig. 2. The cross-section of the pillar is roughly a box-section, with a bead on the inner flange as shown in Fig. 3. The generic pillar is a straight tube with a constant cross section in Fig. 3 (a) from its bottom up to a height of 750 mm. Then, it tapers and curves into the cross section in Fig. 3 (b) at a total height of 1300 mm. The thickness of the external sheet metal (lower halves of cross sections in Fig. 3) is assumed to be fixed at 0.8 mm, whereas the thickness of the internal sheet metal (upper halves of cross sections in Fig. 3) is considered as a design variable chosen from standard gage thicknesses. Additionally, *one* of the three types of reinforcement in Fig. 4 is placed inside the pillar, with two different sheet thicknesses at upper and lower portions, as illustrated in Fig. 5.

The objective of the B-pillar design problem is the minimization of the weight, subject to a prescribed load carrying capacity. As illustrated in Fig. 5, the design variables are the heights of the entire reinforcement and of its lower portion (2 continuous variables), the type of the reinforcements (1 discrete variable among choices in Fig. 4, for both upper and lower portions), and the thicknesses of the upper and lower portions of the reinforcement and the internal sheet of the pillar

(3 discrete variables). Available sheet thicknesses are 0.01mm discrete values between 1.0 mm and 3.2 mm in steps of 0.2 mm, as well as 0.01 mm, which represents the absence of sheet metal in the FE model. In addition, the heights of both upper and lower reinforcements are not allowed to go below 20mm, due to the minimum element size of the FE model. It should be noted that a more accurate FE model of the B-pillar should not include any elements in the reinforcement zone when the thickness of the reinforcement nears zero. The model adopted in this paper is less accurate but presents an easier task to the mesh generator.

The B-pillar is assumed to be fully fixed at the bottom, while its top end is free to deform under the loading. Loading of the pillar is achieved by quasi-static crushing (0.1 m/s) vertically downwards at the top end, and the reaction force of the pillar is recorded. The desired load carrying capacity of the pillar is 4500 kg, without exceeding a deformation of 120 mm. It is noted that although the dimensions and load values presented in this paper are not those of any particular vehicle model, they represent typical values of a compact to a mid-sized car.

An important underlying assumption in this model is that the side-crash requirements are already satisfied within the given search range of the design variables and that the B-pillar is the main load carrying member during roof-crush. This allows for ignoring the simulation of side-crash, and also allows singling out of the B-pillar from the complete side frame of the vehicle. Thus, the size of the FE model is significantly reduced, with too much sacrifice of the accuracy.

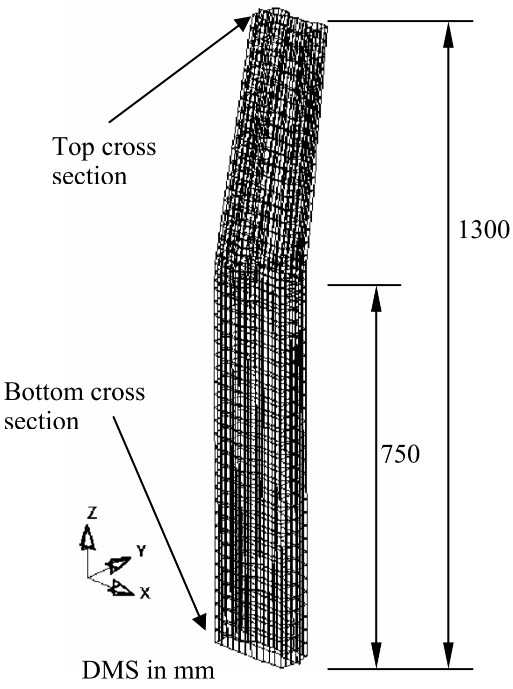

Fig. 2. Finite element model of a generic B-Pillar, viewed from the exterior of the vehicle.

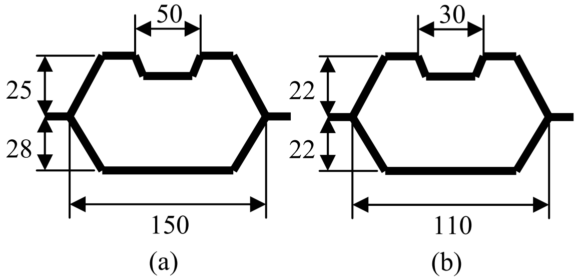

Fig. 3. Dimensions of the cross sections of a generic B-pillar. (a) bottom cross section and (b) top cross section. Upper sides with a bead face to the interior of the vehicle.

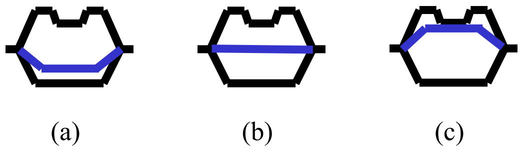

Fig. 4. Reinforcement types for generic B-Pillar. (a) type 0, (b) type 1, and (c) type 2.

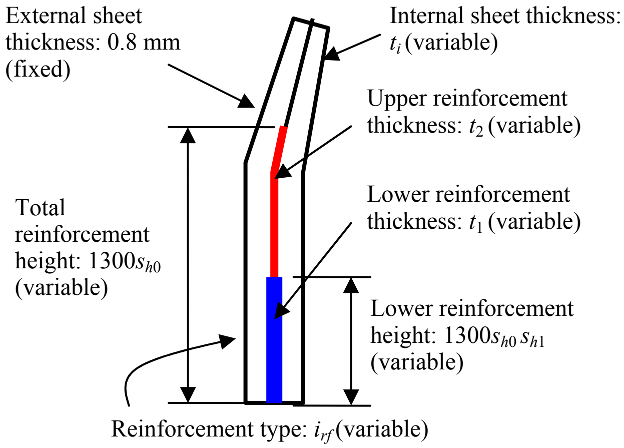
External sheet thickness: 0.8 mm (fixed)
Internal sheet thickness: t_i (variable)
Upper reinforcement thickness: t_2 (variable)
Lower reinforcement thickness: t_1 (variable)
Total reinforcement height: $1300s_{h0}$ (variable)
Lower reinforcement height: $1300s_{h0}s_{h1}$ (variable)
Reinforcement type: i_{rf} (variable)

Fig. 5. Illustration of design variables. Total reinforcement height is given as a fraction s_{h0} of the total height (1300 mm), and lower reinforcement height as a fraction s_{h1} of s_{h0}.

In summary, the optimization model is given as:

$$\text{minimize } f = \text{weight of structure} \tag{1}$$

subject to

$$g_1:\ -P_{\max} + 4500 \le 0 \tag{2}$$

$$g_2:\ -1300s_{h0} + 1300s_{h0}s_{h1} + 20 \le 0 \tag{3}$$

$$g_3:\ -1300s_{h0}s_{h1} + 20 \le 0 \tag{4}$$

$$i_{rf} \in \{0,1,2\}$$

$$s_{h0}, s_{h1} \in [0,1] \tag{5}$$

$$t_i, t_1, t_2 \in \{0.01,\ 1.0, 1.2, 1.4, \ldots, 2.8, 3.0, 3.2\}$$

where

- $P_{\max}$: peak crushing force of the B-Pillar [kg]
- i_{rf} : reinforcement type
- s_{h0} : fraction of reinforcement height to total pillar height (1300 mm)
- s_{h1} : fraction of lower reinforcement height to s_{h0}
- t_i : thickness of internal sheet [mm]
- t_1, t_2 : thickness of upper and lower reinforcements [mm]

Simulation of the B-pillar is performed using a nonlinear finite element model (FEM) using the commercial code LS-DYNA [21]. Prior to solving the above optimization model, several roof crush simulations were conducted with different forces. Fig 6 (solid line) shows a typical force response of the B-pillar FE model. Since the maximum crushing force occurs early on, the simulation stops at 0.05s during the optimization, which we concluded is sufficiently long to capture the occurrence of the peak force (as seen in Fig. 6). Simulating 0.05 seconds of crash event takes approximately 130 seconds on a 3.0GHz PC with 1.0GB RAM. Thus in order to simulate the full crush distance of 125mm, one would require crash event simulation up to 1.25s, which would take about twenty five time the computational time.

Also observed is the existence of noise in the force response curve. To reduce the undesired effect of noise on the optimization, a second order Butterworth digital low pass filter [22] is used to eliminate the high frequency noise. The equation of the filter (for a cutoff-frequency at 10% of a sampling frequency of 2.0 kHz) is given by:

$$P_k = 0.075 f_k + 1.682 \times P_{k-1} - 0.757 \times P_{k-2} \qquad (6)$$

where P_k and f_k are the k-th filtered value and sampled value, respectively. The filtered force curve, shown in a dotted line in Fig. 6, is much smoother than the original with only a slight time delay while capturing the peak force value accurately.

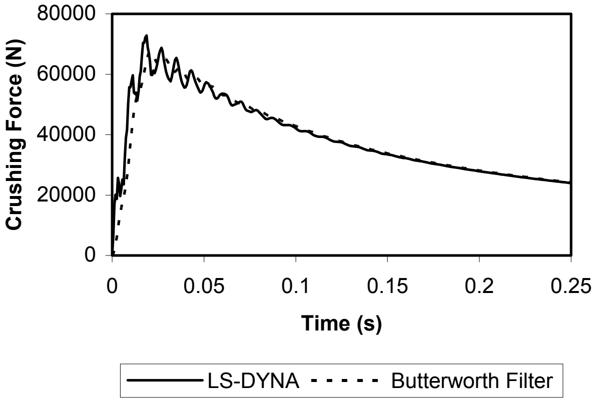

Fig. 6. A typical crushing force response of B-Pillar simulated by LS-DYNA.

Fig 6 show a typical Von Mises stress distribution from an FE simulation, where three zones of high stress are observed: near the loading point at the top, near the point of curvature change in the middle, and near the base of the pillar where it is attached to the other underbody members. Good designs, therefore, may favor reinforcing both zones 2 and 3, only zone 3, or none, depending on the trade-off between the increased weight and stiffness.

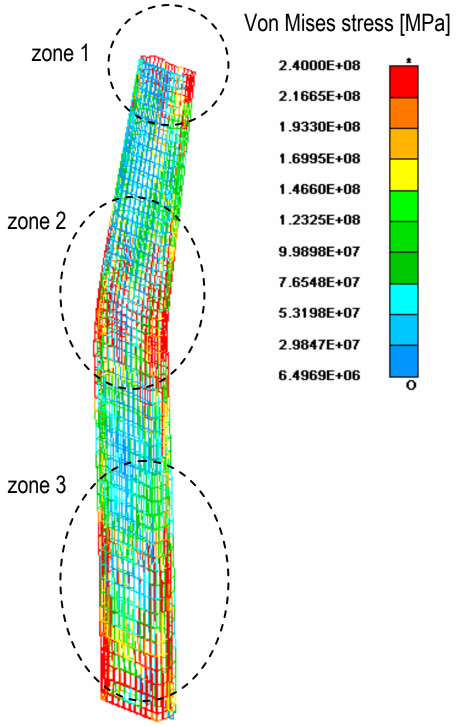

Fig. 9 Typical Von Misses stress distribution in B-Pillar subject to roof crush.

MIXED REACTIVE TABOO SEARCH

Reactive Taboo Search (RTS) is a global optimization algorithm for discrete optimization problems [1]. Its origin, Taboo Search [23-26], is a neighborhood search algorithm where the search direction is biased by prohibiting (*tabooing*) the move to certain points in the search space. Reactive taboo search [1] adaptively varies the way points are tabooed based on the optimization history. An attractive feature of RTS is its efficiency in finding the global optimal with a relatively small number of objective function evaluations [23-26] compared to population-based search algorithms such as Genetic Algorithm [27]. A continuous version of RTS, Continuous Reactive Taboo Search (CRTS), was developed by Battiti and Tecchiolli [3], which employs binary branching over the ranges of continuous variables, followed by local optimization with the Affine Shaker algorithm, and achieved impressive performances on benchmark problems [3]. Using SQP as the local optimizer, CRTS was also applied to surface fitting problems [28]. RTS was extended to integer variables with arbitrary finite ranges and effectively applied to structural optimization [29].

This paper presents a new blend of RTS and CRTS, termed as the Mixed Reactive Taboo Search (MRTS), which is capable of handling mixed integer and continuous variable problems. The basic flow of MRTS follows the one for CRTS [3]. A summary of the steps is presented below, where *P* is a set of the points visited prior to the current iteration, *O* is the local optimum solutions found prior to the current iteration. Each point in *P* is associated with a variable *taboo*, which can take non-negative integer values. A point is called "tabooed" if *taboo* of the point is positive. By returning set *O* at termination, MRTS can find multiple local optima, rather than a single one.

1. (initialization) Evaluate initial point $\boldsymbol{x}_0$, add $\boldsymbol{x}_0$ to *P*, and $\boldsymbol{x} \leftarrow \boldsymbol{x}_0$.
2. (neighborhood search) Evaluate all points in the neighborhood of $\boldsymbol{x}$ and add them in *P*. if a non-tabooed point $\boldsymbol{y}$ better than $\boldsymbol{x}$ is found, $\boldsymbol{x} \leftarrow \boldsymbol{y}$, and go to step 5.
3. (local optimization) if $\boldsymbol{x}$ is not in *O*, run local search (e.g. SQP) starting from $\boldsymbol{x}$ with respect to the continuous variables, and obtain the solution $\boldsymbol{y}$. If *O* contains another local optimum $\boldsymbol{z}$ in the neighborhood of $\boldsymbol{x}$, shrink the neighborhood with respect to the continuous variables. Add the solution $\boldsymbol{y}$ in *O*, $\boldsymbol{x} \leftarrow \boldsymbol{y}$, and go to step 5.
4. (move from a local optimum $\boldsymbol{x}$) if not all points in the neighborhood of x are tabooed, find best non-tabooed point $\boldsymbol{y}$ and $\boldsymbol{x} \leftarrow \boldsymbol{y}$.
5. (update taboo values) Set *taboo* of $\boldsymbol{x}$ as *taboo length* and decrement *taboo* for all points in *P*. If $\boldsymbol{x}$ is in *P*, increment *taboo length*, *cycle count* and reset *no-cycle count*. Otherwise add $\boldsymbol{x}$ in *P*, reset *cycle count*, and increment *no-cycle count*.
6. (shorten taboo period) If *no-cycle count* is sufficiently large, decrement *taboo length*.
7. (quick escape) If *cycle count* is sufficiently large, randomly find a point y not in *P*, $\boldsymbol{x} \leftarrow \boldsymbol{y}$, and reset *cycle count*.
8. (termination or iteration) if termination conditions are satisfied, return *O*. Otherwise go to step 2.

The descriptions for the key steps are provided as:

- **Neighborhood evaluation (step 2):** The neighborhood of a point $\boldsymbol{x}$ is defined as a set of points that differs from $\boldsymbol{x}$ *only* in one variable, by ±1 for the integer variables and by ± *zone size* for the continuous variables. The *zone size* is typically a fraction of the ranges of the continuous variables. Therefore, a neighborhood can contain as many points as the twice the number of variables. However, not all of these points need to be evaluated since MRTS stores all points evaluated in the past.
- **Running local optimizer (step 3):** When none of the neighboring design points are better than the current point, a local optimizer (SQP is used in the current implementation) is run to fine tune the values of the continuous variables to the local optimum. It is to be noted, that if current point is already marked as a local optimum, there is no need to run local optimizer, and the algorithm jumps directly to the next best neighboring design point.
- **Intensifying the search (step 3):** If several distinct local optima are identified in a neighborhood, it is likely that the neighborhood is too large with respect to the continuous variables and thus it is decreased for that region of the search space.
- **Diversifying the search (steps 5 and 6):** To avoid the multiple re-visits to the same local optimal points, MRTS taboos the recently visited points. This tabooing help the search move away from already explored region and find unexplored region of the search space.
- **Quick escape (step 7):** When the algorithm keeps cycling within a region of the search space, this can be detected by the increase of *cycle count*. In such cases, it is likely that tabooing is insufficient to direct the search to a new direction, and a *quick escape* is performed by randomly changing the current point. A quick escape can be seen as a random restart of the search with past memory of exploration.
- **Termination (step 8):** As is the case of many heuristic search algorithms, there are no theoretical criteria to determine the number of iterations necessary to obtain the global optimum. Therefore, the algorithm is typically terminated after a pre-specified number of objective function evaluations.

To illustrate the working of MRTS, consider the following simple mixed integer programming problem:

$$\text{minimize} \quad f = -x_1^2 - x_2^2 \tag{7}$$

subject to

$$g_1 : x_1 \le 4 \tag{8}$$

$$g_2 : -x_1 \le 4 \tag{9}$$

$$g_3 : x_2 \le 4 \tag{10}$$

$$g_4 : -x_2 \le 4 \tag{11}$$

$$g_5 : -x_1 + x_2 \le 4 \tag{12}$$

$$g_6 : x_1 - x_2 \le 4 \tag{13}$$

$$\begin{aligned} &-5 \le x_1 \le 5;\ x_1 \in \mathbf{Z} \\ &-4 \le x_2 \le 4;\ x_2 \in \mathbf{R} \end{aligned} \tag{14}$$

The feasible domain of and the objective function the problem are shown in Figs. 7 and 8, respectively. The problem has two global optima at (4, 4) and (-4, -4) with $f = -32$. With the initial point of (0, 0), *taboo length* = 5, and *zone size* for *x*2 of 2.0, a sample run of MRTS on the above problem can be outlined as follows:

- The neighbors of (0, 0) are (1, 0) and (-1, 0), and for example, (0, 1.3) and (0, -0.8). The last two points are obtained by randomly sampling x_2 within the range [-2.0, 2.0]. Among these four points, the best point in the neighborhood is (0, 1.3) with $f = -1.69$, which is better than the current point (0, 0). Set *taboo* of (0, 0) to be 5, then move to (0, 1.3).

- Similar steps lead to point (0, 2.9), while recording and updating *taboo* of all points previously visited.
- The neighboring points of (0, 2.9) are (-1, 2.9) and (1, 2.9), and for example, (0, 1.5) and (0, 4.1). Although (0, 4.1) has the best value of *f*, it is outside the feasible domain, so the best points in the neighborhood are (-1, 2.9) and (1, 2.9). A tie breaking rule (always use first point among equivalent candidates) decides to move to (-1, 2.9).

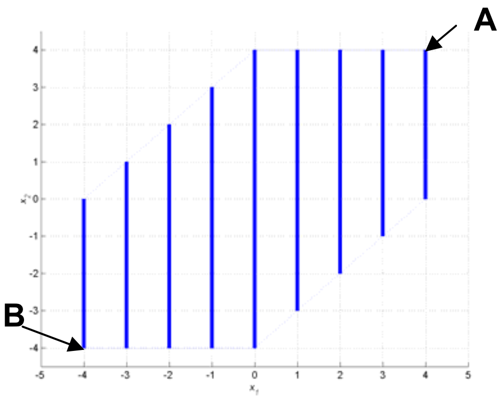

Fig. 7. Feasible space of example problem. Points A (4,4) and B (-4, -4) are both global optimal solution with f = -32.

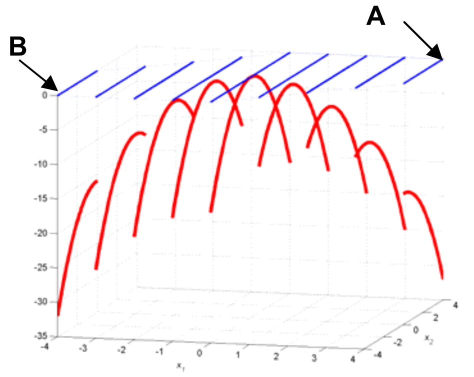

Fig. 8. Objective function of example problem. Note the function has local optimum with respect to x2 at for each value of x1. Global optimal solutions A and B in Fig. 5 are also shown.

- Since the neighbors of (-1, 2.9) are either infeasible or worse, the local optimizer runs for x_2 starting with 2.9, while keeping x_1 fixed to -1. It stops at a local optimal (-1, 3). Record this point, and then move to the next best non-tabooed point in the neighborhood of (-1, 3), which is (0, 3).
- The neighboring points of (0, 3) are (-1, 3) and (1, 3), and for example, (0, 1.8) and (0, 4.6). The point (0, 4.6) is outside the bounds of x_2 so simply *clipped* to become (0, 4.5). However, (0, 4.5) is infeasible. The next best point in the neighborhood, (-1, 3) is tabooed as it was just visited at the previous iteration. So move to the next best point, which is (1, 3).
- Since the current point (1, 3) is near a basin of new attraction, and a series moves along x_1 leads to point (4, 3). Then, the local optimizer reaches to one the global optima of (4, 4).
- Given enough number of iterations (not done in this *hand-simulation* example), the diversification mechanism or the quick escape will direct the search to another basin of attraction, which will eventually lead to another global optimum at (-4, -4).

B-PILLAR OPTIMIZATION RESULTS

The B-pillar roof crush model discussed in the previous section is optimized with Sequential Quadratic Programming (SQP) and MRTS, in order to demonstrate the following advantage of MRTS: 1) global search with relatively small number of function evaluations, 2) identification of multiple local optima with one optimization run, 3) consistent performance regardless of the choice of starting points.

Since SQP is a local optimizer that can handle only continuous variables, the results are obtained using a multi-start strategy with thicknesses t_1, t_2, and t_3 relaxed to the continuous range of [0.01, 3.2]. The resulting optimal solutions are then round to the nearest feasible discrete values in Equation (5). Since there is no obvious way to relax i_{rf} that specifies the reinforcement type, multiple runs are performed in each of the reinforcement type. Table 1 shows the 15 starting points used for the SQP runs and Table 2 lists the top five distinct solutions starting from these points. Table 3 shows the top five solutions round to the nearest discrete values. After the round-off, all solutions become worse (weight increased) and the solutions #1 and #3 become infeasible.

Next, one run of MRTS was conducted starting from the point #1 in Table 1, which resulted in the top five solutions in Table 4. It should be noted that *all* five solutions by MRTS are better than the best discrete solution in Table 3 and most of the relaxed solutions (which use infeasible sheet thicknesses) in Table 2 by SQP. As seen in Tables 3 and 4, some solutions by SQP (solutions 1, 3, and 5) have no reinforcements, while all solutions by MRTS has some reinforcements, all of type 0.

Table 1. Fifteen (15) starting points of B-Pillar optimization used for SQP runs.

#	1-3	4-6	7-9	10-12	13-15
i_{rf}	0,1,2	0,1,2	0,1,2	0,1,2	0,1,2
s_{h0}	0.95	0.5	0.05	0.95	0.05
s_{h1}	0.95	0.5	0.05	0.95	0.05
t_i [mm]	3.2	2.0	1.0	1.0	3.2
t_1 [mm]	2.4	1.2	0.1	0.1	2.4
t_2 [mm]	2.4	1.2	0.1	0.1	2.4

Table 2. Top five (distinct) solutions obtained by 15 SQP runs, starting from the points in Table 1.

#	1	2	3	4	5
i_{rf}	1	0	1	2	0
s_{h0}	0.950	0.350	0.890	0.070	0.055
s_{h1}	0.710	0.670	0.170	0.050	0.050
t_i	2.820	2.220	2.790	2.950	3.090
t_1	0.082	2.360	0.030	2.370	0.060
t_2	0.006	1.550	0.027	1.970	0.002
f	8.339	8.477	8.692	8.802	8.840
g_1	-2.68	-85.82	-10.43	-0.409	-7.485

Table 3. The top five SQP solutions in Table 2 rounded to the nearest discrete values. Note solutions #1 and #3 become infeasible after the round-off.

#	1	2	3	4	5
i_{rf}	1	0	1	2	0
s_{h0}	0.950	0.350	0.890	0.070	0.055
s_{h1}	0.710	0.670	0.170	0.050	0.050
t_i	3.0	2.4	2.8	3.0	3.2
t_1	0.01	2.4	0.01	2.4	0.01
t_2	0.01	1.6	0.01	2.0	0.01
f	8.63	8.86	8.18	8.93	9.07
g_1	+261	-2722	+2507	-705	-1320

Table 4. Top five distinct solutions (local optima) identified by MRTS starting from the point #1 in Table 1.

#	1	2	3	4	5
i_{rf}	0	0	0	0	0
s_{h0}	0.24	0.15	0.14	0.14	0.14
s_{h1}	0.05	0.93	0.83	0.84	0.95
t_i	2.6	2.8	2.8	2.8	2.8
t_1	1.2	1.0	1.2	1.2	1.4
t_2	1.0	2.4	2.2	2.0	2.0
f	8.229	8.508	8.563	8.563	8.601
g_1	-0.622	-1.141	-124.5	-38.5	-0.15

Table 5 summarizes the best solutions and statistics of the 15 SQP runs (with subsequent round-off) and one MRTS run. It is clear to see that MRTS has succeeded in finding a better solution using fewer function evaluations than SQP. Due to the subsequent round-off, the solution by SQP becomes severely over-designed as indicated by the value of g_1.

To demonstrate the performance of MRTS is independent of the choice of a starting point, additional four MRTS runs are performed with randomly chosen starting points, terminated at the same number of function evaluations (1200). Table 6 shows the best solutions for each run. All runs consistently found a better solution than the best solution by SQP in Table 5.

Table 5. Best solutions by SQP (after the round-off) and MRTS.

	SQP	MRTS
i_{rf}	0	0
s_{h0}	0.350	0.24
s_{h1}	0.670	0.05
t_i [mm]	2.4	2.6
t_1 [mm]	2.4	1.2
t_2 [mm]	1.6	1.0
f [kq]	8.86	8.229
g_1	-2722	-0.622
# of Moves	–	102
# of Quick Escapes	–	2
# of Local Optimizer Runs	15	15
# of Local Optima Found	14	11
# of Function Evaluations	1649	1200

Table 6. Best solutions of additional MRTS runs with random start points.

	1	2	3	4
i_{rf}	0	0	0	0
s_{h0}	0.386	0.864	0.078	0.590
s_{h1}	0.391	0.282	0.780	0.408
t_i [mm]	2.2	2.6	3.0	2.0
t_1 [mm]	2.2	1.0	0.8	1.8
t_2 [mm]	1.4	0.01	0.8	1.2
f [kq]	8.227	8.235	8.749	8.185
g_1	-12.4	-86.7	-17.3	-154.3
# of Moves	79	97	118	90
# of Quick Escapes	1	1	1	1
# of Loc. Opt. Runs	18	16	24	15
# of Loc. Opt. Found	14	12	19	12
# of Fn Evaluations	1200	1200	1200	1200

DISCUSSION

In the above example, a set of locally optimal solutions identified by a single MRTS run outperformed the ones by SQP with multiple starting points both in the overall solution quality and the number of required simulation runs. It is also shown that the performance of MRTS is consistent regardless of the starting point.

While SQP is known to be one of the most efficient local optimization techniques for constraint optimization problems [30], the following known drawbacks of SQP became evident:

- **Sensitivity to noise in simulation results:** The evaluation of the transient dynamic response of the nonlinear FEM model inevitably involves numerical errors. From the viewpoint of the optimizer, these errors are equivalent to the noise added to the "true" values of objective function and/or or constraints. Although the objective function of the B-pillar optimization problem is the structural weight requiring no FEM simulation, the noise in the minimum crushing force in the constraint, although filtered, is likely to cause the premature termination of SQP.

- **Necessity to have continuous variables:** Due to the use of gradient, SQP it requires the all variables to be continuous. If a problem involves discrete variables, they are typically relaxed to continuous variables, and the obtained solutions are round to the nearest discrete values for the relaxed variables. While this "*relax and round-off*" strategy can yield improved designs, it very often fails to identify true discrete optima. This is demonstrated in the Table 3, where the round-off solutions are worse than those produced by SQP. Moreover, when some variables are impossible, or very difficult to relax (such as i_{rf} in the above example), attempting SQP runs for all possible combination of such discrete variables can be very prohibitively inefficient.

- **Dependency on starting point:** Since SQP is a local optimizer, the ending point of its search will be the nearest stationary point in the basin of attraction of the objective function to the given starting point. Attempts to find global optima of a multi-modal objective function by using different starting points become more and more futile as the number of variables increase.

On the other hand, the results obtained by MRTS are good in several aspects. MRTS operates directly on the problem variables without the need for relaxation of discrete variables. Thus the final obtained solution of MRTS is true discrete optima. MRTS can be thought of as "*a local optimizer that can explore beyond local optima*" with many attractive features of local optimization techniques such as the ability to efficiently explore descent directions in the objective function. For this reason, MRTS is likely to outperform GA [27] for mixed discrete continuous problems, in terms of the number of functions evaluations to reach a solution with a certain quality [29].

A main concern when using MRTS (or any flavor of taboo search) is the termination of the search. Unlike local optimizers which stop upon satisfying local optimality criteria, there is no way to guarantee that a good design point MRTS has found is a global optimum, or that it will find a better one if allowed to run longer. This is an obvious disadvantage, which requires some experience (or experimentation) to determine the appropriate number of function evaluations to allow MRTS to do. On the other hand, being able to specify the maximum number of function evaluations is an advantage, because it allows the selection of such number based on the availability of computational resources, as done in the above case study. Recently, Tian *et al.* [31] proved the asymptotic convergence of Taboo Search based on Markov Chain Analysis. They also showed MRTS is guaranteed to do no worse than a comparative local search. It is expected that further theoretical development will suggest a systematic way to determine the termination criteria.

CONCLUSION

This paper discussed the crashworthiness design of a vehicle B-pillar subjected to roof crush with Mixed Reactive Taboo Search, an efficient global optimization technique for mixed discrete and continuous problems. The technique was effective in finding multiple good solutions with a single run, independent of the choice of the starting design. A single run of MRTS to the B-Pillar design problem identified a set of better designs with smaller number of simulation runs, than multiple runs of Sequential Quadratic Programming (SQP) with several starting points.

ACKNOWLEDGMENTS

This research is funded by Nissan Technical Center North America, Inc. (NTCNA). The authors acknowledge Mr. Masakazu Nakamura at NTCNA for general technical information on vehicle B-pillar structures and the federal regulations for roof crush requirements.

REFERENCES

[1] National Highway Traffic Safety Administration, 2001, *Traffic Safety Facts 2000: A Compilation of Motor Vehicle Crash Data from the Fatality Analysis Reporting System and the General Estimation System*, US Department of Transportation, DOT HS 809 337.

[2] Battiti, R. and Tecchiolli, G., 1994, "The Reactive Tabu Search," *ORSA Journal on Computing*, Vol. 6, pp. 126-140.

[3] Battiti, R. and Tecchiolli, G., 1996, "The Continuous Reactive Tabu Search: Blending Combinatorial Optimization and Stochastic Search for Global Optimization," *Annals of Operations Research*, Vol. 63, pp. 153-188.

[4] Mayer R., Kikuchi, N. and Scott, R., 1996, "Application of Topological Optimization Techniques to Structural Crashworthiness," *International Journal for Numerical Methods in Engineering,* Vol. 39, pp. 1383-1403.

[5] Mayer, R., Maurer, D. and Bottcher, C., 2000, "Application of Topological Optimization Program to the Danner Test Simulation," Proceedings of the *ASME 2000 Design Engineering and Technical Conferences*, September 10-13, Baltimore, Maryland, DETC 2000-DAC 14292.

[6] Luo, J., Gea, C. and Yang, R., 2000, "Topology Optimization for Crush Design," Proceedings of the *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 6-8, Long Beach, CA, AIAA Paper Number: AIAA-2000-4770.

[7] Soto, A., 2001, "Optimal Structural Topology Design for Energy Absorption: A Heuristic Approach," Proceedings of the *ASME 2001 Design Engineering and Technical Conferences*, September 9-12, Pittsburgh, PA, DETC 2001-DAC 21126.

[8] Soto, A., 2001, "Structural Topology for Crashworthiness Design by Matching Plastic Strain and Stress Levels,"

Proceedings of the *ASME 2001 International Mechanical Engineering Congress and Exposition*, November 11-16, New York, NY, IMECE 2001-AMD 25455.

[9] Bennett, J. A., Lust, R. V. and Wang, J.T., 1991, "Optimal Design Strategies in Crashworthiness and Occupant Protection," *ASME Winter Annual Meeting,* Atlanta, GA, AMD - Vol. 126, pp. 51-66.

[10] Soto, C. and Diaz, A., 1999, "Basic Models for Topology Design Optimization in Crashworthiness Problems," Proceedings of the *ASME 1999 Design Engineering and Technical Conference*, September 12-15, Las Vegas, Nevada, DETC 99 / DAC 8591.

[11] Chellappa, S. and Diaz, A., "A Multi-Resolution Reduction Scheme for Structural Design," Proceeding of the *2002 NSF Design, Service and Manufacturing Grantees and Research Conference*, January 2002, pp. 98-107.

[12] Yang, R. J., Tho, C. H., Wu, C. C., Johnson, D. and Cheng, J., 1999, "A Numerical Study of Crash Optimization," Proceedings of the *ASME 1999 Design Engineering and Technical Conferences*, September 12-15, Las Vegas, Nevada, DETC 99 / DAC 8590.

[13] Yang, R. J., Gu, L., Liaw, L., Gearhart, C., Tho, C. H., Liu, X. and Wang, B. P., 2000, "Approximations for Safety Optimization of Large Systems," Proceedings of the *ASME 2000 Design Engineering and Technical Conference*s, September 10-13, Baltimore, Maryland, DETC 2000 / DAC 14245.

[14] Yang, R. J., Wang, N., Tho, C. H., Bobineau, J. P. and Wang, B. P., 2001, "Metamodeling Development for Vehicle Frontal Impact Simulation," Proceedings of the *ASME 2001 Design Engineering and Technical Conference*s, September 9-12, Pittsburgh, PA, DETC 2001 / DAC 21012.

[15] Shi, Q. Hagiwara, I. and Takashima, F., 1999, "The Most Probable Optimal Design Method for Global Optimization," Proceedings of the *ASME 1999 Design Engineering and Technical Conference*s, September 12-15, Las Vegas, Nevada, DETC 99 / DAC 8635.

[16] Redhe, M. and Nilsson, L., 2002, "Using Space Mapping and Surrogate Models to Optimize Vehicle Crashworthiness Design," Proceedings of the *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 4-6, Atlanta, Georgia, AIAA Paper Number: AIAA-2002-5536.

[17] Han, J. and Yamazaki, K., 2000, "Maximization of the Crushing Energy Absorption of the S-Shaped Thin-Walled Square Tube," Proc. of the *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 6-8, Long Beach, CA, AIAA Paper Number: AIAA-2000-4750.

[18] Kurtaran, H., Omar, T. and Eskandarian, A., 2001, "Crashworthiness Design Optimization of Energy-Absorbing Rails for the Automotive Industry," Proceedings of the *ASME 2001 International Mechanical Engineering Congress and Exposition*, November 11-16, New York, NY, IMECE 2001 / AMD 25452.

[19] Chen, S., 2001, "An Approach for Impact Structure Optimization using the Robust Genetic Algorithm," *Finite Elements in Analysis and Design*, Vol. 37, pp. 431-446.

[20] Yang, R. J., Gu, L., Tho, C. H. and Sobieski, J., 2001, "Multidisciplinary Optimization of a Full Vehicle with High Performance Computing," Proceedings of the *2001 American Institute of Aeronautics and Astronautics Conference*, pp. 688-698, AIAA Paper Number: AIAA-2001-1273.

[21] Livermore Software Technology Corporation, 2001, *LS-DYNA user manuals*, Livermore, CA, USA 94550.

[22] Mathworks, Inc., 2000, *MATLAB Signal Processing Toolbox User Guide*.

[23] Glover, F., 1986, "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computers and Operations Research*, Vol. 13, pp. 533-549.

[24] Glover, F., 1989, "Tabu Search – Part I," *ORSA Journal on Computing*, Vol. 1, pp. 190-206.

[25] Glover, F., 1990, "Tabu Search – Part II," *ORSA Journal on Computing*, Vol. 2, pp. 4-32.

[26] Glover, F., 1999, *Tabu Search*, Kluwer Academic Publishers.

[27] Goldberg, D., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Inc.

[28] Youssef, A., 2001*, Reverse Engineering of Geometric Surfaces using Tabu Search Optimization Techniques*, M.Sc. Thesis, Mechanical Design and Production Dept., Cairo University.

[29] Hamza, K., Mahmoud, H. and Saitou, K., 2002, "Design Optimization of N-Shaped Roof Trusses," Proceedings of *GECCO 2002: Genetic and Evolutionary Computations Conference*, New York, July 2002, pp. 1089-1096.

[30] Mathworks, Inc., 2000, *MATLAB Optimization Toolbox User Guide*.

[31] Tian, P., Ma, J. and Fan, Z., 1997, "A Stochastic Tabu Search Strategy and its Global Convergence," Proceedings of the *IEEE International Conference on Systems, Man and Cybernetics*, Piscataway, NJ, Vol. 1, pp. 410-414.